Universidade Federal
do Rio de Janeiro
Escola Politécnica

SISTEMA INTELIGENTE PARA AVALIAÇÃO EM TEMPO REAL DA ATENÇÃO DOS CONDUTORES DE VEÍCULOS

Thiago Koster Lago

Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Miguel Elias M. Campista

Rio de Janeiro
Novembro de 2019

SISTEMA INTELIGENTE PARA AVALIAÇÃO EM TEMPO REAL DA ATENÇÃO DOS CONDUTORES DE VEÍCULOS

Thiago Koster Lago

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRÔNICO E DE COMPUTAÇÃO

Autor:

Thiago Koster Lago

Orientador:

Prof. Miguel Elias Mitre Campista

Examinador:

Prof. José Gabriel Rodríguez Carneiro Gomes

Examinador:

Prof. Heraldo Luis Silveira de Almeida

Rio de Janeiro

Novembro de 2019

Declaração de Autoria e de Direitos

Eu, *Thiago Koster Lago* CPF *141.834.937-22*, autor da monografia *SISTEMA INTELIGENTE PARA AVALIAÇÃO EM TEMPO REAL DA ATENÇÃO DOS CONDUTORES DE VEÍCULOS*, subscrevo para os devidos fins, as seguintes informações:

1. O autor declara que o trabalho apresentado na disciplina de Projeto de Graduação da Escola Politécnica da UFRJ é de sua autoria, sendo original em forma e conteúdo.

2. Excetuam-se do item 1. eventuais transcrições de texto, figuras, tabelas, conceitos e ideias, que identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos proprietários, quando necessárias.

3. O autor permite que a UFRJ, por um prazo indeterminado, efetue em qualquer mídia de divulgação, a publicação do trabalho acadêmico em sua totalidade, ou em parte. Essa autorização não envolve ônus de qualquer natureza à UFRJ, ou aos seus representantes.

4. O autor pode, excepcionalmente, encaminhar à Comissão de Projeto de Graduação, a não divulgação do material, por um prazo máximo de 01 (um) ano, improrrogável, a contar da data de defesa, desde que o pedido seja justificado, e solicitado antecipadamente, por escrito, à Congregação da Escola Politécnica.

5. O autor declara, ainda, ter a capacidade jurídica para a prática do presente ato, assim como ter conhecimento do teor da presente Declaração, estando ciente das sanções e punições legais, no que tange a cópia parcial, ou total, de obra intelectual, o que se configura como violação do direito autoral previsto no Código Penal Brasileiro no art.184 e art.299, bem como na Lei 9.610.

6. O autor é o único responsável pelo conteúdo apresentado nos trabalhos acadêmicos publicados, não cabendo à UFRJ, aos seus representantes, ou ao(s) orientador(es), qualquer responsabilização/ indenização nesse sentido.

7. Por ser verdade, firmo a presente declaração.

Thiago Koster Lago

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Escola Politécnica - Departamento de Eletrônica e de Computação
Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária
Rio de Janeiro - RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es).

DEDICATÓRIA

Dedico este trabalho ao meu avô, que instigou minha curiosidade e sempre me servirá de inspiração.

AGRADECIMENTO

Um imenso agradecimento à minha família que me apoiou durante este meu difícil percurso, principalmente meus pais, que tanto me ajudaram para a minha educação. Um agradecimento à todos os professores da instituição pelos ensinamentos dados. Agradeço também aos meus amigos de curso pelas horas de estudo e por passarmos este período de nossas vidas juntos.

RESUMO

O uso de dispositivos móveis durante a direção de um veículo é um dos principais causadores de acidentes. Portanto, este trabalho tem como objetivo criar um sistema capaz de identificar momentos de desatenção dos motoristas em tempo real, e assim, diminuir o número de acidentes no trânsito através da emissão de alertas. Para isso, algoritmos de visão computacional são usados para determinar a direção do olhar do motorista que, em conjunto com dados do veículo como velocidade e aceleração, inferem se há ou não desatenção. A ideia chave é calibrar a rotação máxima da face do motorista em função da velocidade e da aceleração do veículo. Caso o veículo esteja em baixa aceleração ou baixa velocidade, a rotação pode ser mais acentuada. Além da rotação da face, o fechamento dos olhos é usado como critério para determinação de desatenção. O sistema utiliza um procedimento de janela temporal deslizante para evitar oscilações entre estados de desatenção e atenção. A implementação propôs uma arquitetura de hardware e software composta por dispositivos com boa relação custo-benefício e código aberto. Os resultados experimentais em ambientes controlados e reais se mostraram satisfatórios, detectando com precisão e rapidez o estado de atenção dos motoristas.

Palavras-Chave: Internet das Coisas, visão computacional, monitoração.

ABSTRACT

The use of mobile devices while driving a vehicle is one of the major causes of accidents. Therefore, this work aims to create a system that can identify drivers' moments of inattention in real time, and thus reduce the number of traffic accidents by alerting them. For this, computer vision algorithms are used to determine the direction of the driver's gaze which, together with vehicle data such as speed and acceleration, infer whether or not there is inattention. The key idea is to calibrate the maximum rotation of the driver's face as a function of vehicle speed and acceleration. If the vehicle is at low acceleration or low speed, the rotation may be more pronounced. In addition to face rotation, eye closure are detected and used as a criterion for determining inattention. The system uses a temporal sliding window procedure to prevent oscillations between inattention and attention states. The implementation proposed a hardware and software architecture composed of cost-effective and open source devices. Experimental results in controlled and real environments were satisfactory, accurately and quickly detecting drivers' attention.

Key-words: Internet of Things, computer vision, monitoring.

SIGLAS

IoT - *Internet of Things*

RPC - *Remote Procedure Call*

AMQP - *Advanced Message Queuing Protocol*

Abramet - Associação Brasileira de Medicina do Tráfego

NCSA - *National Center for Statistics and Analysis*

GPS - *Global Positioning System*

OBD - *On Board diagnostics*

CONAMA - Conselho Nacional do Meio Ambiente

HTML - *Hypertext Markup Language*

USB - *Universal Serial Bus*

I2C - *Inter-Integrated Circuit*

GPIO - *General-Purpose Input/Output*

HOG - *Histogram of oriented gradients*

EAR - *Eye Aspect Ratio*

MQTT - *Message Queuing Telemetry Transport*

Sumário

1 Introdução **1**

1.1 Delimitação 3

2 Problemas e Trabalhos Relacionados **4**

2.1 Desafios 4

2.1.1 Estado do motorista 4

2.1.2 Padrões de inferência do olhar 5

2.1.3 Dificuldade da fiscalização 6

2.2 Trabalhos Relacionados 6

3 Arquitetura do sistema **9**

3.1 Arquitetura do hardware 9

3.2 Arquitetura do software 10

3.2.1 Módulo Principal 11

3.2.2 Módulo de Vídeo 12

3.2.3 Módulo de Face 12

3.2.4 Módulo de Severidade 12

3.2.5 Módulo de Recepção 13

3.2.6 Módulo de Publicação 13

3.2.7 Módulo de Log 13

3.2.8 Módulo de Notificação 13

3.2.9 Módulo de Alerta 13

4 Implementação **14**

4.1 Implementação dos módulos 15

4.1.1 Módulo principal 15

4.1.2 Módulo de Vídeo 17
4.1.3 Módulo de Log 18
4.1.4 Módulo de Severidade 18
4.1.5 Módulo de Face 21
4.1.6 Módulo de Recepção 26
4.1.7 Módulo de Publicação 29
4.1.8 Módulo de Notificação 31
4.1.9 Módulo de Alerta 31

5 Experimentos e Resultados **33**
5.1 Testes preliminares 33
5.1.1 GPS 33
5.1.2 Acelerômetro 34
5.2 Testes do Módulo de Face 35
5.2.1 Sem integração 36
5.2.2 Com integração 37
5.3 Teste em campo 42

6 Conclusão **47**
6.1 Conclusão 47
6.2 Trabalhos futuros 48

Bibliografia **49**

A Cálculo dos ângulos **53**
A.1 Calculo dos ângulos 53

B Informações complementares **56**
B.1 GPS 56

C Configurações **57**
C.1 Configuração do Raspberry Pi 57

Lista de Figuras

3.1 Representação do sistema com a fonte de energia, os dados dos sensores como entrada e os arquivos produzidos como saída. 10
3.2 Diagrama de comunicação entre os módulos do sistema. 11

4.1 Portas GPIO do Raspberry Pi 3 em destaque pelo retângulo vermelho. 14
4.2 Diagrama do hardware. 15
4.3 Fluxo principal 16
4.5 Leitura do *buffer* para contorno do problema de baixo desempenho para o fluxo do arquivo de vídeo. 18
4.6 Fluxo da webcam com leitura direta do dispositivo. 18
4.7 Curvas obtidas pelo Módulo de Severidade. 19
4.8 Superfícies obtidas pelo Módulo de Severidade através das curvas representadas pelas Figuras 4.7a e 4.7b. 20
4.9 Fluxo do Módulo de Severidade. 21
4.10 Fluxograma do Módulo de Face. 22
4.11 Ilustração dos eixos de rotação da face. 23
4.12 Diagrama de transformação. 24
4.13 Processo de obtenção das coordenadas no espaço. 25
4.14 Imagens dos pontos utilizados. 26
4.15 Fluxograma da obtenção dos dados dos sensores. 27
4.16 Diagrama de comunicação RPC. 29
4.17 Conexão entre Raspberry PI e acelerômetro. 30
4.18 Saída do comando i2cdetect. 30
4.19 Fluxograma de comunicação entre os módulos de Notificação e Alerta. 32

5.1 Resultados do teste do GPS. 34

5.2 Configurações para o teste do acelerômetro. 35
5.3 Gráfico dos valores de Pitch e Yaw ao longo do tempo e distração detectada. 36
5.4 Teste do módulo de face utilizando parâmetros padrões. 37
5.5 Parâmetros utilizados durante o teste. 38
5.6 Gráfico dos valores de Pitch e Yaw ao longo do tempo. 39
5.7 Teste do módulo de face com integração de velocidade. 40
5.8 Superfícies utilizadas no teste. 40
5.9 Gráfico dos valores de Pitch e Yaw ao longo do tempo com influência da velocidade e aceleração. 41
5.10 Olhando para direita com influência da velocidade e acelerômetro. . . 42
5.11 Rota registrada pelo sistema. Inicio na marcação verde e fim na vermelha. 42
5.12 Resultado dos ângulos de rotação ao longo do tempo. 43
5.13 Valores e eye aspect ratio medidos pelo sistema. 44
5.14 44
5.15 Sistema detectando uma falsa distração. 45
5.16 45

A.1 Ilustração dos sistemas de coordenadas utilizados. 53

C.1 Menu de configuração. 57

Lista de Tabelas

4.1 Tabela dos registradores do MPU-6050. 31

5.1 Resultado das medições para a configuração A e B. 35

B.1 Tabela do protocolo utilizado no GPS 56

Capítulo 1

Introdução

A popularização dos dispositivos eletrônicos móveis, como os smartphones, e o aumento da conectividade em rede [1], torna comum que motoristas se distraiam com alguma aplicações, mesmo enquanto guiam os veículos. O uso dos smartphones, portanto, divide a atenção e a concentração do motorista, impactando drasticamente no tempo de resposta a eventos inesperados e, consequentemente, nas chances de ocorrer um acidente [2]. De acordo com a Associação Brasileira de Medicina do Tráfego, a Abramet, o uso do celular ao volante já é a terceira maior causa de mortes no trânsito brasileiro [3]. Nos Estados Unidos, 9% dos 34,247 acidentes ocorridos em 2017 tiveram como causa a distração do motorista, sendo responsável por 3,166 mortes de acordo com o estudo realizado pela NCSA (*National Center for Statistics and Analysis*) [4].

Uma alternativa para inibir o uso destes aparelhos durante a direção é o desenvolvimento de sistemas capazes de detectar a desatenção dos condutores enquanto dirigem. Para isso, é necessário o uso de algoritmos de processamento de imagem em conjunto com dados obtidos do veículo e do ambiente ao redor.

Este projeto visa inferir a desatenção do usuário a partir da detecção da direção do olhar do motorista em conjunto com dados de posição e velocidade dos veículos. A ideia é verificar se a direção do olhar em uma determinada região da cidade pode ser considerada segura ou não, levando em conta a velocidade do veículo.

O sistema a ser desenvolvido deve fornecer uma decisão confiável e de forma rápida. Para isso, a qualidade da câmera, dos sensores e do computador no qual a

avaliação é realizada devem ser consideradas durante o projeto.

O sistema a ser desenvolvido tem como objetivo a detecção de momentos de desatenção do usuário com base na direção do olhar. Portanto, o projeto proposto deve implementar uma aplicação capaz de: (1) identificar rostos, olhos e características faciais necessárias; (2) detectar a direção do olhar; (3) obter dados complementares: velocidade, luminosidade, e (4) classificar e alertar se o motorista está desatento com bases nas informações obtidas em tempo real.

Portanto, o trabalho proposto difere dos trabalhos relacionados apresentados no Capítulo 2 pois além de utilizar dados de diferentes fontes, faz sua análise em tempo real e alerta o usuário em momentos de distração.

O sistema proposto tem como componente principal um módulo central de decisão. Este módulo recebe informações provenientes de GPS, microfone e sensores em geral para definir em tempo real os parâmetros a serem utilizados na classificação do estado de atenção do motorista. Dessa forma, é realizado um estudo inicial para a utilização de microcontroladores na obtenção de dados dos sensores, assim como uma pesquisa para definir uma arquitetura de comunicação eficiente entre o computador principal e o microcontrolador responsável pela aquisição de dados. Para estimar a direção do olhar da face do usuário, é aplicada uma técnica geométrica, que consiste na localização de pontos de interesse no rosto e a realização de cálculos para estimar os ângulos de rotação nos eixos longitudinal e transversal. Esta técnica possui uma simples implementação e apenas necessita a utilização de uma câmera. A aplicação utiliza uma câmera para obter pontos de interesses na face do usuário através de algoritmos de processamento de imagem. A partir dos pontos obtidos é possível inferir a direção do olhar do motorista. Combinando os dados obtidos com os parâmetros gerados pelo módulo de decisão, é feita uma análise do estado da atenção do motorista. Caso o estado seja de desatenção, um alerta é gerado para o motorista. Além do alerta, todos os dados e inferências são armazenados em um arquivo de registro para auditoria *offline*. O hardware depois de testado em bancada pode ser instalado no painel de um veículo para a verificação de seu funcionamento em um ambiente real.

O projeto foi analisado com testes em bancada e em campo. Os testes em bancada tiveram como objetivo verificar o funcionamento individual e integrado de

cada módulo enquanto que os testes em campo validar o desempenho do sistema no seu ambiente de operação normal. O sistema projetado foi capaz de identificar momentos de desatenção dos motoristas durante os experimentos em campo porém apresentou piora no desempenho de acordo com a falta de luminosidade no ambiente. Este problema pode ser solucionado com a utilização de um iluminador de infravermelho.

1.1 Delimitação

O Capítulo 2 apresenta os trabalhos relacionados ao projeto proposto e os problemas a serem solucionados. O Capítulo 3 introduz a arquitetura do sistema proposto. Já a implementação do projeto e a descrição dos softwares utilizados são realizadas no Capítulo 4. O Capítulo 5 contém os experimentos, em ambiente controlado e em cenários reais, e os seus respectivos resultados. Por fim, o Capítulo 6 conclui o projeto e aponta direções futuras.

Capítulo 2

Problemas e Trabalhos Relacionados

Este capítulo apresenta os desafios existentes no tema deste projeto, assim como os trabalhos relacionados e as respectivas abordagens para solução dos problemas.

2.1 Desafios

O desafio a ser solucionado converge na análise do estado do motorista. Dentre vários fatores, um que torna possível determinar em que condição o condutor se encontra é a identificação da direção do olhar. A identificação da direção do olhar do condutor facilita a fiscalização ou prevenção de acidentes, já que pode tanto ser usada como uma forma de inferir a atenção do motorista quanto uma forma de alertá-lo sobre possível desatenção. A seguir são apresentados os principais fatores que contribuem com esta análise.

2.1.1 Estado do motorista

O estado em que se encontra o motorista é um grande influenciador no desempenho durante a condução de um veículo, já que influencia o tempo de reação do indivíduo. A direção pode ser prejudicada por distrações, realização de múltiplas tarefas, cansaço e sonolência. De acordo com a NHTSA (*National Highway Traffic Safety Administration*), a distração ocorre quando o motorista desvia sua atenção

da tarefa de dirigir para uma outra atividade. A atividade em foco neste trabalho é o uso do celular. Porém, existem outras fontes comuns de distração como comer, conversar com passageiros ou mexer no painel do veículo. Um acidente tem sua causa classificada por distração quando é um acidente em que o motorista estava distraído no momento da colisão [5].

A NHTSA classifica distrações nas seguintes categorias: visuais (motorista desviando olhar da estrada), auditivas (reagir a um celular tocando), biomecânicas (ajustar o rádio) e cognitivas (estar perdido em pensamentos) [6]. Além de distrações, a sonolência é um grande fator no impacto do tempo de reação de um motorista. Um estudo realizado na França analisa o desempenho de motoristas privados e não privados de sono [7]. Os motoristas dirigiram 1000 quilômetros durante 10 horas em sessões de 105 minutos. O estudo concluiu que a duração da atividade não é o principal fator que prejudica a direção. O tempo acordado e a duração do sono, porém, são os que possuem um impacto mais significativo. A fadiga combinada com a sonolência afetam de forma significativa o tempo de reação do motorista.

Os principais fabricantes já demonstraram interesse em desenvolver sistemas com o propósito de detectar o estado do motorista. Nos carros mais recentes estão presentes sistemas como: (1) *Rest recommendation system* da Audi [8]; (2) *Active driving assistant* da BMW [9]; (3) *Fatigue detection system* da Volkswagen [10]; e (4) *Driver alert* da Ford [11]. Todos esses sistemas citados realizam a mesma função que é alertar o motorista caso ele seja detectado em estado de desatenção ou sonolência. Como exemplo, o sistema *Rest recommendation system* da Audi analisa os movimentos do volante para detectar se a atenção do motorista está diminuindo. Caso o motorista esteja desatento, um indicador no painel é acendido além de emitir um sinal sonoro para alertar o usuário [8].

2.1.2 Padrões de inferência do olhar

Inferir a direção do rosto de uma pessoa é um problema recorrente no campo de visão computacional e é abordado em aplicações de diferentes áreas, como por exemplo, para análise psicológica [12] e na segurança do trabalho de construtores [13]. As principais formas para a detecção da direção do olhar de uma pessoa são: (1) utilizar a direção do rosto; (2) rastrear o movimento dos olhos; e (3) uti-

lizar uma combinação das duas anteriores. Lex Fridman et al. [14], por exemplo, concluíram que a adição da detecção do movimento dos olhos aumenta, em média, a precisão da detecção da direção do olhar em 5,4%. Portanto, a detecção da direção do olhar no sistema proposto é feita utilizando apenas o movimento do rosto do motorista, porque esse aumento da precisão não justifica a complexidade que a detecção dos olhos iria adicionar ao sistema.

2.1.3 Dificuldade da fiscalização

De acordo com Código de Trânsito Brasileiro Art. 252, inciso V e VI utilizar um aparelho de celular durante a direção de um veículo é considerado uma infração. Porém segundo a Abramet, o uso do celular ao volante já é a terceira maior causa de mortes no trânsito brasileiro [3]. A razão de não existir uma coleta consistente de dados sobre as causas de acidentes de trânsito, em particular quando o motivo é o celular, é a atribuída à difícil fiscalização desse tipo de infração. Essa dificuldade é oriunda da falta de evidências do seu uso, ao contrário de infrações como dirigir embriagado, velocidade excessiva ou a não utilização de cinto de segurança, por exemplo [15].

2.2 Trabalhos Relacionados

Em virtude da variedade de fontes de distração apresentada na seção anterior, há diferentes campos de estudo para sistemas que têm como objetivo a detecção de distração de motoristas, sendo os principais:

- Fisiológico: Tem como foco o estado do motorista e, como consequência, observa dados como batimentos cardíacos, atividade cerebral e abertura dos olhos para detecção de sonolência.

- Comportamental: Tem como foco o estado do veículo e, como consequência, utiliza dados do veículo como frenagem, aceleração e curvas bruscas para detectar momentos de desatenção do usuário.

- Visual: Tem como foco a obtenção de informações do interior e exterior do veículo através do uso de câmeras e, como consequência, pode identificar o uso

de celular, motorista com foco fora da estrada, avanço de semáforo e veículo saindo da faixa, por exemplo.

- Híbrido: Realiza uma composição das categorias anteriores.

Wang et al. [16] utilizam dados provenientes de eletroencefalografia durante a operação do veículo para detectar se o motorista está realizando uma outra tarefa além de dirigir e se, portanto, está desatento.

Este é um exemplo de estudo fisiológico para a detecção da atenção do motorista. O projeto apresentou uma precisão de 90%, mas necessita de equipamentos intrusivos e pode apresentar dificuldade de aceitação entre os usuários.

Um exemplo de estudo comportamental é o aplicativo proposto por Khandakar et al. [17]. Este aplicativo monitora dados do veículo e controla o uso do celular. O sistema monitora o comportamento do usuário como: o uso do celular, frenagem e mudança de faixa abruptas. Esse sistema, porém, apenas monitora e restringe a utilização do celular e não alerta o usuário em caso de distração. Ainda, Khandakar et al [17] usam equipamentos em conformidade com o padrão OBD-2 *(On-board diagnostics)* para obtenção de dados e diagnóstico dos veículos. No Brasil, esse tipo de padrão pode não estar disponível em alguns dos veículos em circulação nas rodovias brasileiras, já que o padrão nacional OBDBr-2 é visto como mais promissor. Segundo o CONAMA, o OBDBr-2 se tornou disponível para a totalidade de veículos leves de passageiros e comerciais, produzidos ou importados para o mercado interno, a partir de 1° de janeiro de 2011 [18].

O sistema proposto por Lamia e Hoque [19] faz o uso do ângulo da face e dos olhos para determinar a desatenção do usuário. Portanto, se trata de um exemplo de estudo visual. O sistema classifica o usuário como desatento se o ângulo da direção do olhar do motorista ultrapassa um limite fixo por um determinado período de tempo. Nessa direção, os autores atingem uma precisão de 92% para a detecção de distração (desvio do olhar). Esse método, apesar de eficiente, limita o movimento do usuário, independente do contexto em que se situa o veículo.

O sistema de Craye et al. [20] utiliza os campos fisiológico, comportamental e visual. Portanto, se trata de um estudo híbrido.

Ele utiliza batimentos cardíacos, posição dos pedais e volante, áudio e vídeo para determinar a desatenção do usuário em um simulador. Portanto, ele possui

abrangência em todas as áreas, fisiológica, comportamental e visual. Outra característica interessante é a escolha da arquitetura modular. No artigo, os autores citam a utilização de três módulos independentes, o visual, o auditivo e o de sinais. Com esse sistema híbrido, os autores foram capazes de atingir uma precisão para detecção de fadiga e distração de 98,4% e 90,5%, respectivamente. Apesar de possuir uma excelente precisão para ambas as detecções, o sistema depende de sensores invasivos que podem causar desconforto ao motorista (oxímetro CMS-50E). Além disso, o sistema não analisa a desatenção e fadiga em tempo real. Os autores explicam que deixaram esta funcionalidade para trabalhos futuros.

Conclui-se que existem diferentes campos de atuação no que se diz respeito a detecção de desatenção de um motorista. Além disso, os projetos apresentados como exemplo neste capítulo pertencem a um ou mais destes diferentes campos. O projeto proposto neste trabalho se enquadra nos campos visual e comportamental e fisiológico, portanto se trata de um estudo híbrido. A inferência da desatenção é feita em tempo real de modo que seja possível alertar o motorista desatento e a análise visual é complementada pelos dados comportamentais como velocidade e aceleração do veículo por exemplo.

Capítulo 3

Arquitetura do sistema

Este capítulo apresenta em alto nível a arquitetura do sistema em termos de hardware e software, deixando detalhes de implementação e decisões de projeto para o Capítulo 4.

3.1 Arquitetura do hardware

O sistema proposto deve possuir poder computacional suficiente para processar imagens em tempo real. Também é necessário exigir fácil instalação e mobilidade para que o sistema seja instalado em um automóvel sem prejudicar a operação do veículo. Para ser possível contextualizar o estado do veículo, é necessário receber dados provenientes de outros sensores além de uma câmera, como por exemplo, GPS e acelerômetro. A saída do sistema é um arquivo de *log*, um arquivo *HTML* contendo a rota do veículo e, caso desejado, um arquivo de vídeo com os quadros processados pelo sistema. Portanto, o sistema deve possuir armazenamento persistente para permitir a escrita desses três tipos de arquivos. A utilização de uma tela para visualização da saída de vídeo do sistema é opcional. Outra necessidade é possuir uma fonte de energia para alimentar o sistema. Na Figura 3.1, é possível visualizar a representação geral do sistema proposto.

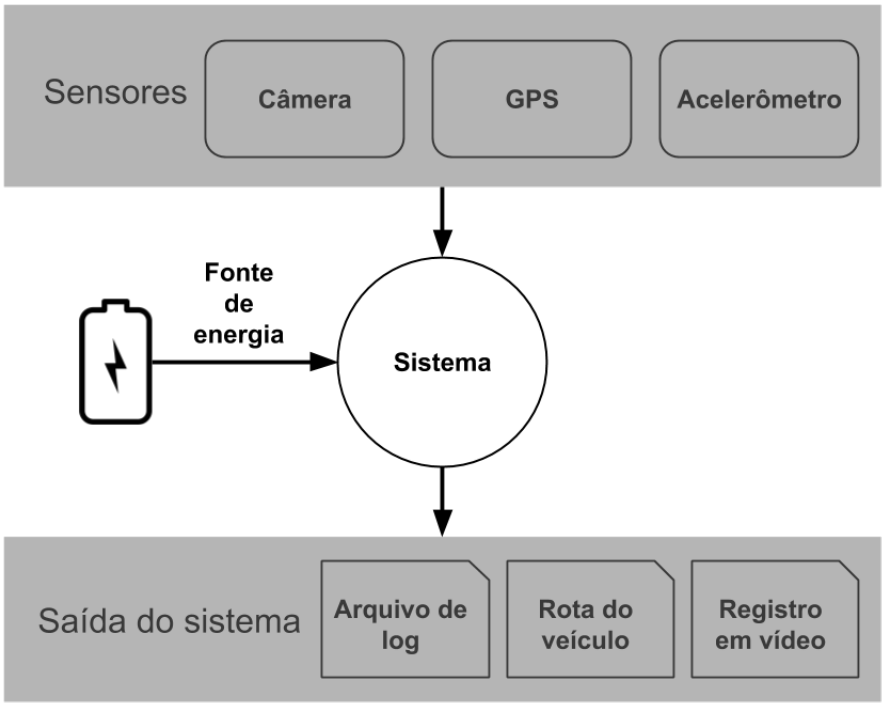


Figura 3.1: Representação do sistema com a fonte de energia, os dados dos sensores como entrada e os arquivos produzidos como saída.

3.2 Arquitetura do software

Esta seção apresenta cada módulo do sistema proposto, como visto na seção anterior, e suas principais funcionalidades.

O projeto do sistema é baseado em uma arquitetura modular com a finalidade de manter o código organizado e de manutenção, depuração, reuso e alteração simples. Sendo assim, a programação do sistema pode ser separada em módulos autocontidos que implementam diferentes funcionalidades. O sistema contém um total de nove módulos: o Módulo Principal e os módulos auxiliares de Vídeo, Face, Severidade, Recepção, Publicação, *Log*, Notificação e Alerta. Devido à assincronia do Módulo de *Log*, Vídeo e Severidade, ambos são implementados em *threads* diferentes da principal. O sistema possui ainda um arquivo de configuração no qual o usuário pode determinar parâmetros que serão utilizados pelos módulos. A Figura 3.2 exemplifica a relação entre cada modulo do sistema, os módulos configuráveis estão marcados conforme a legenda e as cores representam as diferentes

threads em que os módulos são instanciados. A linha tracejada representa a comunicação entre módulos que podem estar instalados em diferentes dispositivos. Já a linha contínua representa a comunicação de módulos implementados no mesmo dispositivo. A descrição individual de cada módulo é introduzida a seguir.

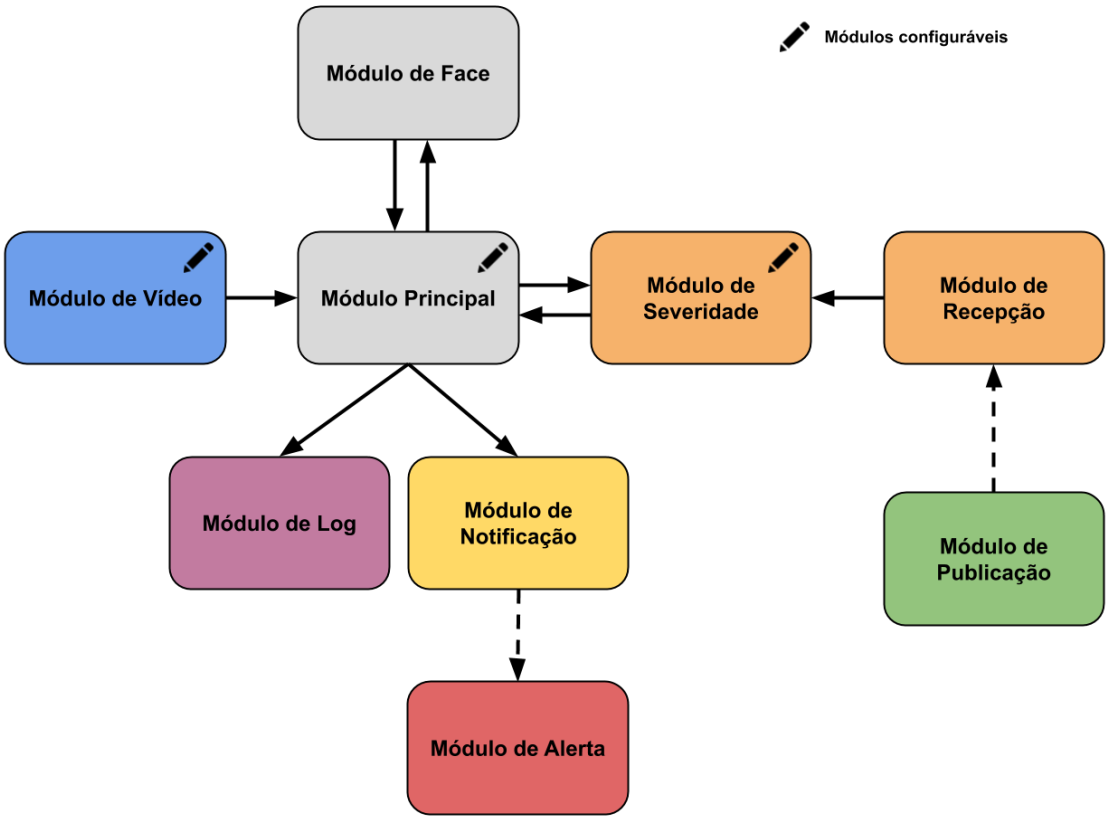


Figura 3.2: Diagrama de comunicação entre os módulos do sistema.

3.2.1 Módulo Principal

Este módulo tem como principal função a orquestração dos módulos auxiliares. Portanto, ele executa o fluxo principal do programa, transfere informações de um módulo para outro conforme a Figura 3.2 e também possui a responsabilidade da inicialização do sistema. Através do arquivo de configuração, é possível determinar se a saída do sistema deve ser armazenada de forma persistente e, caso positivo, deve exibir informações como ângulos do rosto, velocidade do veículo dos quadros do vídeo. Por fim, ao término da execução do sistema, o módulo principal traça a rota do veículo em um mapa e registra o resultado em um arquivo *html* para ser visto posteriormente.

3.2.2 Módulo de Vídeo

O sistema, além de receber o vídeo de uma câmera em tempo real, deve também ser capaz de analisar um vídeo previamente gravado. Essa funcionalidade adicional é implementada para atribuir ao sistema a capacidade de auditoria. Dessa forma, tanto um vídeo gravado por uma fonte diferente quanto um vídeo gravado pelo próprio sistema pode ser analisado externamente ao veículo, por exemplo. Portanto, o Módulo de Vídeo através de um parâmetro configurável determina o tipo de entrada de vídeo.

3.2.3 Módulo de Face

Este módulo tem como função estimar a direção do rosto do motorista. Para isso, ele recebe pontos de interesse da face e realiza cálculos para determinar os ângulos de rotação do rosto. A direção do olhar do motorista é inferida caso o ângulo limite de um eixo de rotação, determinado pelo Módulo de Severidade (Seção 3.2.4), seja extrapolado. Outra funcionalidade é determinar se os olhos do motorista estão fechados utilizando os pontos de interesse. Caso o motorista esteja com os olhos fechados ou olhando para alguma direção que não seja para frente, este módulo determina que o motorista está distraído.

3.2.4 Módulo de Severidade

O Módulo de Severidade correlaciona o comportamento do usuário com a sua atenção ao conduzir o veículo. Pode-se dizer, portanto, que o Módulo de Severidade é responsável pela inteligência do sistema. Por exemplo, um motorista ao dirigir a uma velocidade baixa tem um ângulo máximo permitido de rotação da face maior que um motorista que dirige em altas velocidades. Conforme a velocidade aumenta, o ângulo permitido diminui já que o motorista deve manter maior atenção ao trânsito. Os parâmetros utilizados pelo Módulo de Face são gerados a partir do Módulo de Severidade. Sendo assim, este módulo tem como principal funcionalidade a contextualização do estado de atenção do motorista, utilizando dados provenientes do Módulo de Recepção (Seção 3.2.5), e fornecimento dos limites toleráveis ao Módulo de Face.

3.2.5 Módulo de Recepção

Este módulo é instanciado pelo Módulo de Severidade e tem como função receber a mensagem que contém os dados provenientes dos sensores. Os dados contidos na mensagem são encaminhados para o Módulo de Severidade.

3.2.6 Módulo de Publicação

Este módulo representa a interface entre o sistema e os sensores. O Módulo de Publicação é responsável por receber os dados dos sensores e criar a mensagem a ser enviada ao Módulo de Recepção com as informações obtidas. Este módulo, por exemplo, obtém dados do GPS e acelerômetro nos experimentos realizados.

3.2.7 Módulo de Log

Este módulo possui a função de registrar o estado do veículo, como velocidade, posição e aceleração. Além disto, o Módulo de *Log* possui também a função de registrar o estado do motorista, como posição inferida de olhar, se os olhos estão fechados e se o motorista está distraído, por exemplo. O registro é realizado em um intervalo de tempo configurável e é criado um arquivo de texto com todas as informações para uma avaliação posterior.

3.2.8 Módulo de Notificação

O Módulo de Notificação tem como função o envio do estado do motorista, distraído ou atento, para o Módulo de Alerta.

3.2.9 Módulo de Alerta

O Módulo de Alerta recebe o estado do motorista através do Módulo de Notificação. Caso o estado do motorista seja de distração, este módulo deve alertá-lo. A importância do alerta é tornar possível mudar o atual estado do motorista e evitar um possível acidente.

Capítulo 4

Implementação

Este capítulo apresenta a implementação dos módulos definidos no sistema proposto, assim como os equipamentos e as principais tecnologias utilizadas. Este projeto requer a utilização de um notebook e um Raspberry Pi para atender seus requisitos. O notebook foi escolhido para realizar o papel de computador principal devido à sua mobilidade. O Raspberry Pi é usado por possuir uma linha de pinos GPIO *(General-Purpose Input/Output)*, como pode ser observado pelo destaque na Figura 4.1. O Raspberry Pi é utilizado como interface entre os sensores e o notebook. Já a comunicação entre o notebook e o Raspberry Pi é realizada de forma assíncrona, utilizando o protocolo *Advanced Message Queuing Protocol* AMQP 0-9-1 e arquitetura de comunicação *Remote Procedure Call* (RPC) que é apresentada com mais detalhe na Seção 4.1.6. O vídeo é fornecido para o notebook com a utilização de uma *webcam*. A Figura 4.2 mostra a implementação da arquitetura no nível de hardware do sistema proposto.

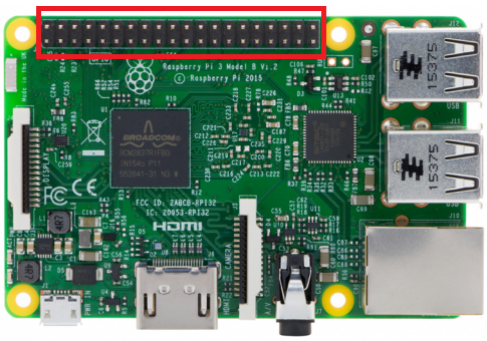


Figura 4.1: Portas GPIO do Raspberry Pi 3 em destaque pelo retângulo vermelho.

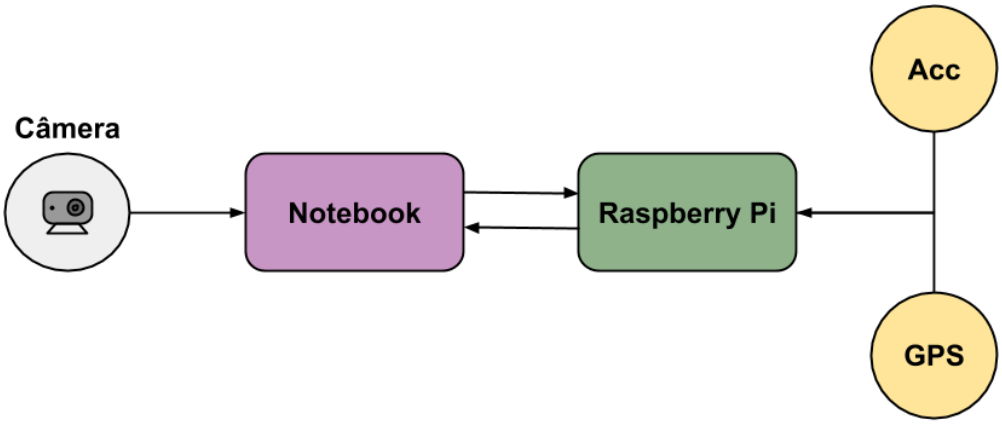


Figura 4.2: Diagrama do hardware.

O Módulo de Publicação, como visto na Seção 3.2.6, exige uma conexão direta com os sensores para a obtenção dos dados. Portanto, ele é executado pelo Raspberry Pi. A implementação deste e dos demais módulos serão apresentadas na próxima seção.

4.1 Implementação dos módulos

Esta seção tem como objetivo apresentar como cada módulo descrito no Capítulo 3 foi desenvolvido, as tecnologias utilizadas e as decisões de implementação tomadas durante o projeto.

4.1.1 Módulo principal

Este módulo lê o arquivo de configuração, inicializa o sistema e inicia o fluxo principal. Este fluxo é demonstrado na Figura 4.3. O módulo recebe, através do método `read()`, e trata o quadro proveniente do Módulo de Vídeo. Depois, aplica a detecção de rosto e pontos faciais implementados pela biblioteca Dlib, vista em mais detalhe na Seção 4.1.1.1. Os dados obtidos são enviados ao Módulo de Face através do método `shape_to_face()`. Após isto, o Módulo Principal obtém os limites dos ângulos de rotação calculados pelo Módulo de Severidade com os dados dos sensores (Seção 4.1.4) através do método `get_all_params()` e os envia ao Módulo de Face com o método `update_params()`. Com isso, é possível obter a direção estimada pelo Módulo de Face para o rosto do usuário através do método `get_face_directions()`

e o estado de atenção do motorista com o método `get_driver_state()`. Por fim, o Módulo Principal obtém os dados dos sensores e envia as informações necessárias ao Módulo de Log com o método `set_status()` e o estado de atenção para o Módulo de Notificação utilizando o método `set_distracted()`.

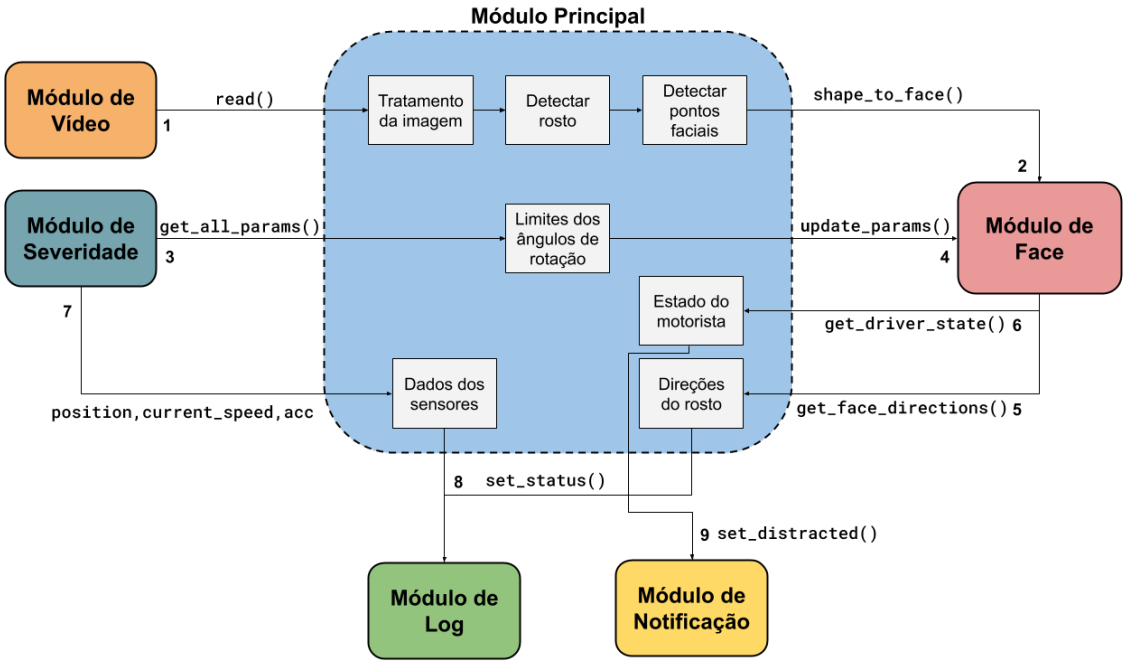
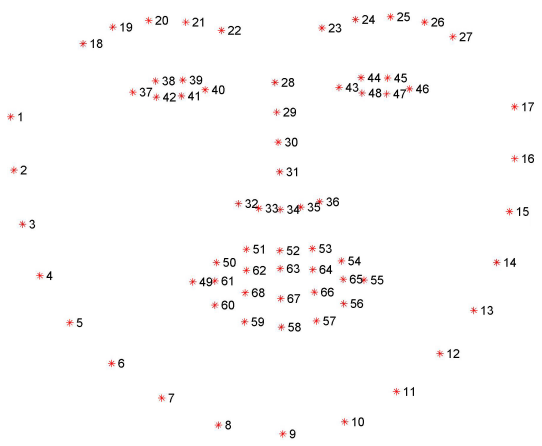


Figura 4.3: Fluxo principal

4.1.1.1 Biblioteca Dlib

O detector de face da biblioteca Dlib [21] usa o recurso de Histograma de Gradientes Orientados (HOG) combinado com um classificador linear, uma pirâmide de imagem e um esquema de detecção de janela deslizante. O estimador padrão de pontos faciais da biblioteca Dlib [22] foi previamente treinado com o dataset IBUG 300-W [23]. Esse dataset contém 50 vídeos em que cada quadro tem uma marcação dos pontos faciais. O modelo estima 68 pontos faciais, representados como visto na Figura 4.4a. Desses 68 pontos, foram selecionados 14 para a utilização do sistema, mostrados na Figura 4.4b. O motivo da quantidade menor de pontos será explicado na Seção 3.2.3.



(a) Representação dos 68 pontos faciais.

(b) Aplicação da biblioteca Dlib para captura de pontos de interesse.

4.1.2 Módulo de Vídeo

A biblioteca OpenCV (*Open Source Computer Vision Library*) [24] é utilizada para capturar a entrada de vídeo. Essa biblioteca possui interfaces para diversas linguagens de programação e suporte para os principais sistemas operacionais. A OpenCV também foi utilizada para calcular os ângulos da face do usuário utilizando os dados obtidos pela biblioteca Dlib [21], como pode ser visto na Seção 4.1.5.

O sistema proposto deve ser capaz de usar como entrada um vídeo recebido através de uma câmera ou um vídeo recebido através de arquivo previamente gravado. Cada uma das opções disponíveis exige uma implementação particular. Ao ler arquivos de vídeo previamente gravados, a biblioteca OpenCV realiza operações que bloqueiam a I/O do sistema. Nota-se como consequência uma grande diminuição de desempenho ao utilizar arquivos de vídeo para análise no sistema. Para contornar esse problema, a classe `FileVideoStream` foi implementada. Essa classe é executada em uma *thread* diferente da principal e mantém preenchido um *buffer* com os quadros do vídeo. Dessa forma, através do método `read()`, o módulo retorna um quadro do *buffer* e não diretamente do arquivo. Esse fluxo é exemplificado pela Figura 4.5. Caso a *webcam* seja utilizada, os quadros podem ser obtidos diretamente do dispositivo, como visto na Figura 4.6.

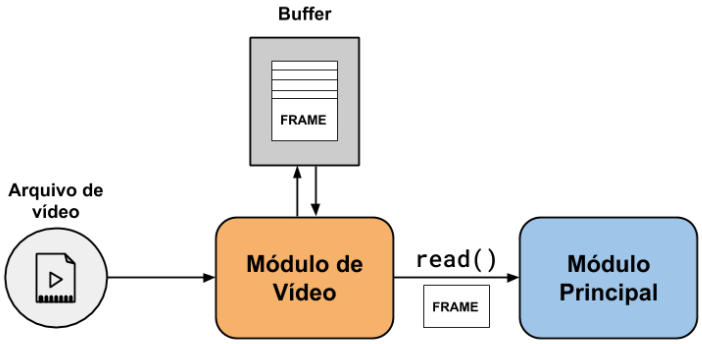


Figura 4.5: Leitura do *buffer* para contorno do problema de baixo desempenho para o fluxo do arquivo de vídeo.

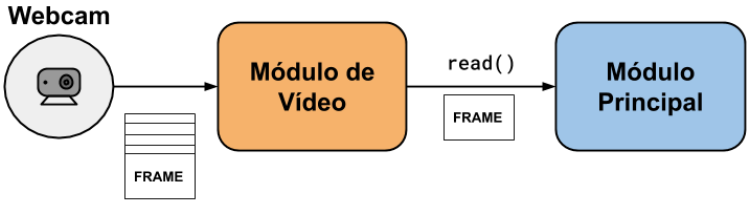


Figura 4.6: Fluxo da webcam com leitura direta do dispositivo.

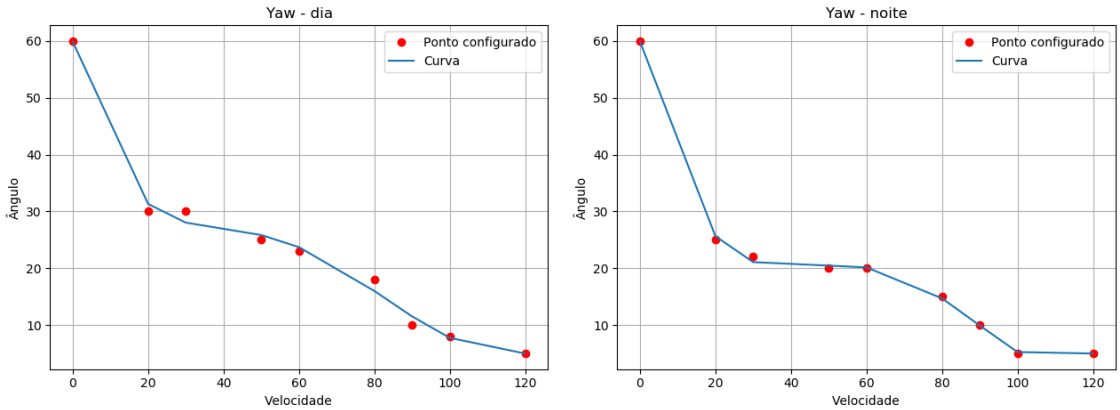
4.1.3 Módulo de Log

O Módulo de Log registra em um intervalo de tempo configurável o estado atual do motorista e veículo. Portanto, deve ser implementado de forma assíncrona para não interferir no fluxo principal do sistema. Para isso, ele foi implementado na classe `LogThread`. A classe `LogThread` é instanciada, iniciada e recebe a mensagem a ser escrita no arquivo de log pelo Módulo Principal.

4.1.4 Módulo de Severidade

Este módulo centraliza as informações provenientes dos sensores no Raspberry Pi e determina os parâmetros que serão utilizados como limites no Módulo de Face, apresentado na Seção 4.1.5. O Módulo de Severidade em sua inicialização obtém as curvas de limite para o eixo transversal e longitudinal. Cada eixo possui

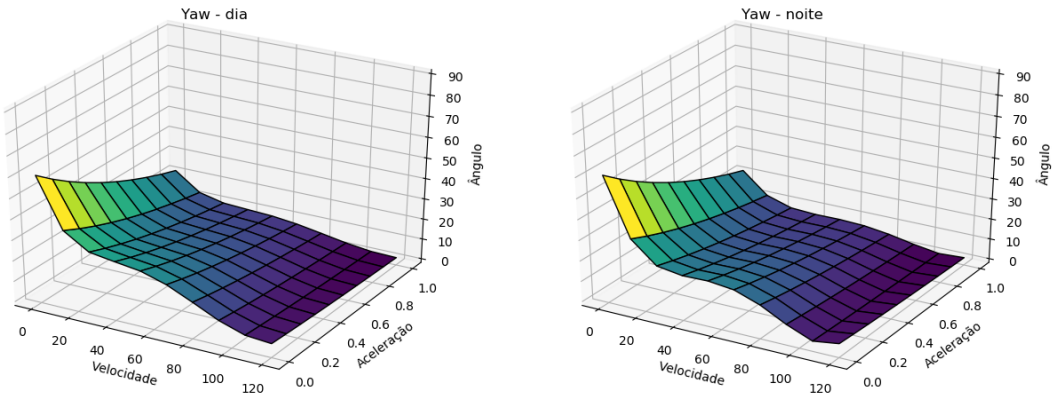
uma curva para o período diurno e outra para o noturno. As curvas são obtidas através do ajuste de um polinômio de quinto grau com os pontos definidos no arquivo de configuração, um exemplo de curva obtida pode ser visto através das Figuras 4.7a e 4.7b. A curva descreve o ângulo máximo permitido no respectivo eixo em função da velocidade do veículo. Ressalta-se que, no período noturno, os ângulos máximos são menores. Isso ocorre pois assume-se que a atenção no trânsito deve aumentar, como consequência da provável redução da luminosidade.



(a) Curva dos limites no período diurno. (b) Curva dos limites no período noturno.

Figura 4.7: Curvas obtidas pelo Módulo de Severidade.

A partir das curvas, é criada uma superfície que descreve o ângulo máximo permitido em função da velocidade e da aceleração do veículo. Um decaimento exponencial foi utilizado para gerar os pontos no eixo da aceleração. Desta forma, o ângulo máximo tem um impacto significativo com o aumento da aceleração mesmo em velocidades baixas. A vantagem da utilização do método apresentado para a geração dos limites é a ausência da necessidade de treinamento do sistema antes de sua utilização. O sistema pode ser configurado com os parâmetros no arquivo de configuração e ser ajustado de acordo com a necessidade de cada usuário. As Figuras 4.8a e 4.8b exemplificam as superfícies geradas através das Figuras 4.7a e 4.7b. O valor do decaimento exponencial foi obtido empiricamente durante os testes em bancada.



(a) Superfície dos limites no período diurno. (b) Superfície dos limites no período noturno.

Figura 4.8: Superfícies obtidas pelo Módulo de Severidade através das curvas representadas pelas Figuras 4.7a e 4.7b.

O Módulo de Severidade inicializa uma instância do Módulo de Recepção para obter os dados atualizados dos sensores. Os dados são obtidos com a chamada RPC, que é realizada através do método `sensors_call()`. Esse procedimento é repetido em um intervalo de tempo definido anteriormente no arquivo de configuração.

O Módulo de Severidade utiliza a biblioteca astronômica *skyfield* [25]. Com essa biblioteca, é possível determinar o horário em que o sol nasce e se põe no local atual com as coordenadas da posição do veículo obtidas com o GPS. O período do dia é determinado comparando o horário do sistema com o horário do pôr do sol calculado. Após essa determinação, é possível calcular as superfícies a serem utilizadas. Além disso, com a obtenção dos dados dos sensores e as superfícies, os limites são definidos e enviados ao Módulo Principal, que por sua vez transmite ao Módulo de Face, como visto anteriormente na Seção 4.1.1. Caso não seja possível obter algum dado do sensor por erro na conexão ou nos sensores, o Módulo de Severidade utiliza valores padrão. O fluxo seguido pelo Módulo de Severidade pode ser visto através da Figura 4.9.

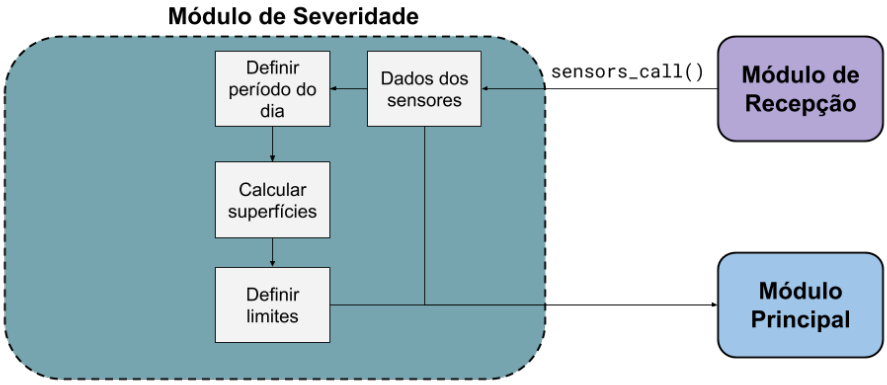


Figura 4.9: Fluxo do Módulo de Severidade.

4.1.5 Módulo de Face

Este módulo possui elevada complexidade, pois é responsável por todas as funções referentes à inferência da direção do rosto e fechamento dos olhos do motorista. A direção é determinada através do estimativa dos ângulos de rotação longitudinal e transversal da face, vista em maiores detalhes na Seção 4.1.5.1. O fechamento dos olhos é determinado pelo método de cálculo de *Eye Aspect Ratio* (EAR) [26], apresentado em mais detalhe na Seção 4.1.5.2. Caso o motorista esteja com os olhos fechados ou não esteja olhando para a frente é atribuído o estado de desatento. Este estado será obtido pelo Módulo Principal e enviado ao Módulo de Alerta 4.1.9 através do Módulo de Notificação 4.1.8.

O funcionamento do módulo segue o seguinte fluxo: O Módulo de Severidade (Seção 4.1.4) envia através da função `shape_to_face()` os pontos faciais a serem utilizados. Após o recebimento, a matriz de rotação é calculada. Em seguida, calcula-se o EAR e os ângulos de rotação no eixo transversal e longitudinal. O cálculo da matriz e a obtenção dos ângulos podem ser vistos em mais detalhes na Seção 4.1.5.1. O Módulo de Severidade faz a atualização dos parâmetros utilizados como limites através do método `update_params()`. De posse destes parâmetros e dos ângulos de rotação, é possível atualizar os contadores que serão utilizados para o processo de atribuição da direção do olhar. Este processo é explicado na Seção 4.1.5.1. Em seguida, o Módulo Principal obtém a direção do rosto inferida utilizando o método `get_face_directions()` e se o motorista está com os olhos

fechados através do método `get_eyes_closes()`. Por fim, o estado do motorista (desatento ou atento) é obtido com o método `get_driver_state()`. O fluxo descrito é apresentado na Figura 4.10.

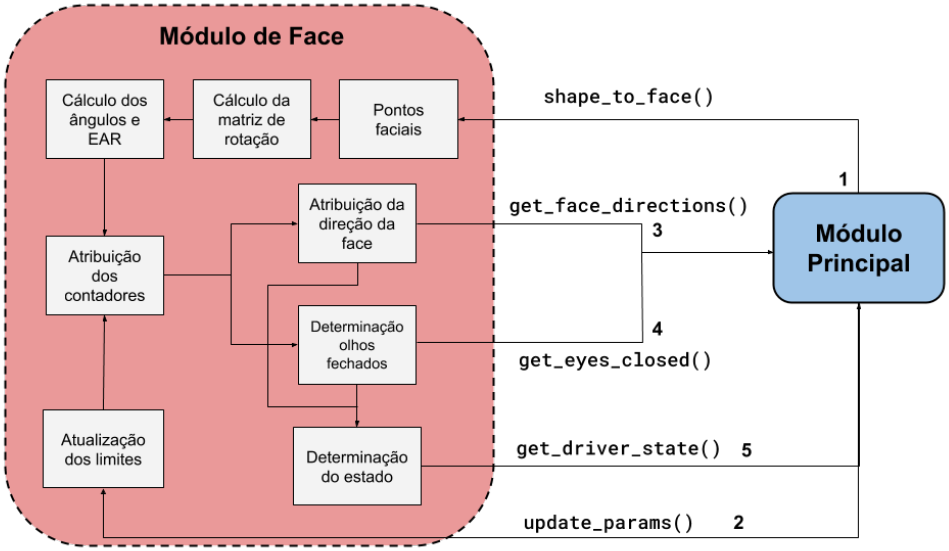


Figura 4.10: Fluxograma do Módulo de Face.

4.1.5.1 Determinação da direção do olhar

Os ângulos de interesse no projeto são os de rotação nos eixos longitudinal (*yaw*) e transversal (*pitch*). A Figura 4.11 exemplifica a localização dos eixos citados. O eixo sagital (*roll*) não é utilizado no projeto por não ser uma rotação comum para o rosto dos motoristas e não ser correspondente à uma distração.

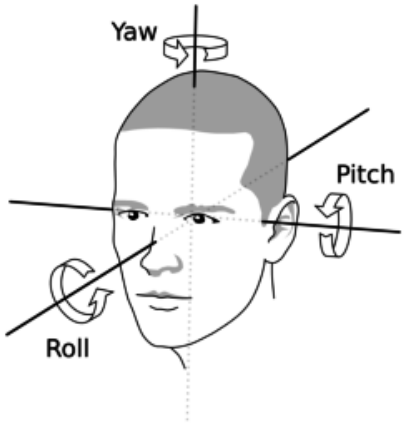


Figura 4.11: Ilustração dos eixos de rotação da face.

Para o cálculo da pose da face, três sistemas de coordenadas são utilizados. As coordenadas do espaço, da câmera e da imagem. Descobrindo a rotação e translado (**R,t**), é possível transformar um ponto nas coordenadas no espaço em um ponto no sistema de coordenadas da câmera. Estes pontos são representados pelos pontos P e p, respectivamente, na Figura 4.12. Em seguida, o ponto da câmera pode ser projetado na imagem utilizando os parâmetros intrínsecos da câmera. Esse processo pode ser visto através na Figura 4.12. Utilizando os pontos obtidos pelo modelo 3D e os pontos calculados pela Dlib, é possível obter os ângulos de rotação da face através do processo descrito no Apêndice A.1.

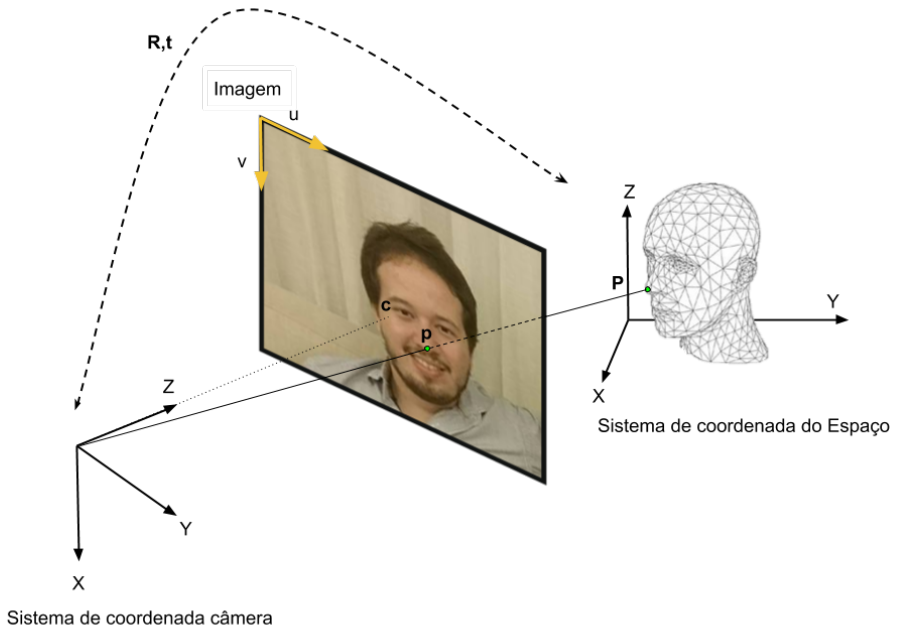


Figura 4.12: Diagrama de transformação.

Portanto, para determinar os ângulos de rotação nos eixos, as seguintes informações são necessárias.

- **Coordenada dos pontos de interesse na imagem:** São os pontos de interesse da face, detectados pela Dlib.
- **Coordenada dos pontos de interesse no espaço:** Pontos 3D em posições fixas para os pontos de interesse utilizados.
- **Parâmetros da câmera:** Distância focal da câmera, o centro óptico na imagem e os parâmetros de distorção radial. Estes parâmetros são aproximados neste trabalho para que a calibração da câmera não seja necessária.

Para obter os pontos de interesse facial utilizados pelo Módulo de Face no sistema de coordenadas no espaço, utilizou-se um modelo 3D de um rosto humano genérico, disponível para *download* no *website TurboSquid* [27]. O nariz foi atribuído como origem do sistema e, em seguida, os pontos de interesse foram identificados no modelo e as coordenadas foram obtidas manualmente. O processo de obtenção das coordenadas foi realizado com o *software* de modelagem 3D Blender [28], que pode ser visto na Figura 4.13.

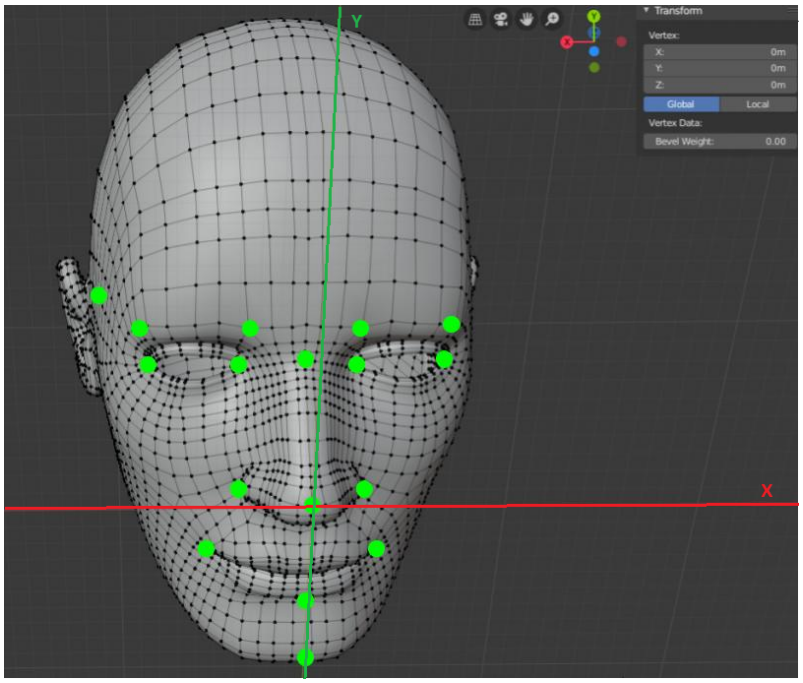


Figura 4.13: Processo de obtenção das coordenadas no espaço.

O Apêndice A.1 apresenta um vetor com os pontos obtidos através do modelo 3D.

De posse dos ângulos da face, são utilizados dois tipos de limites para a inferência do olhar do motorista: limites de valores e de contadores. O limite de valor é utilizado como ângulo máximo permitido para cada eixo. Já o de contador determina a quantidade máxima de quadros consecutivos em que o limite de valor pode ser extrapolado sem que a respectiva direção seja atribuída. Quando o ângulo calculado pelo sistema para um determinado quadro ultrapassa o limite de valor determinado pelo Módulo de Severidade, o contador referente à direção extrapolada é acrescentado. Caso o angulo seja menor que o limite, o respectivo contador é subtraído. O sistema determina se a face está orientada para uma determinada direção verificando se algum contador passou por um determinado limite de contadores. A implementação deste conceito de contadores foi necessária para dar ao sistema mais suavidade na inferência da direção do olhar e fechamento dos olhos. Desta forma, o estado atual determinado pelo sistema fica mais estável e menos sujeito a variações rápidas. Outro benefício desta implementação é a possibilidade de modificação dos limites utilizados.

4.1.5.2 Cálculo de *Eye Aspect Ratio*

O parâmetro *Eye Aspect Ratio* (EAR) [26] é calculado utilizando um total de seis pontos para cada olho do usuário, conforme a Figura 4.14.

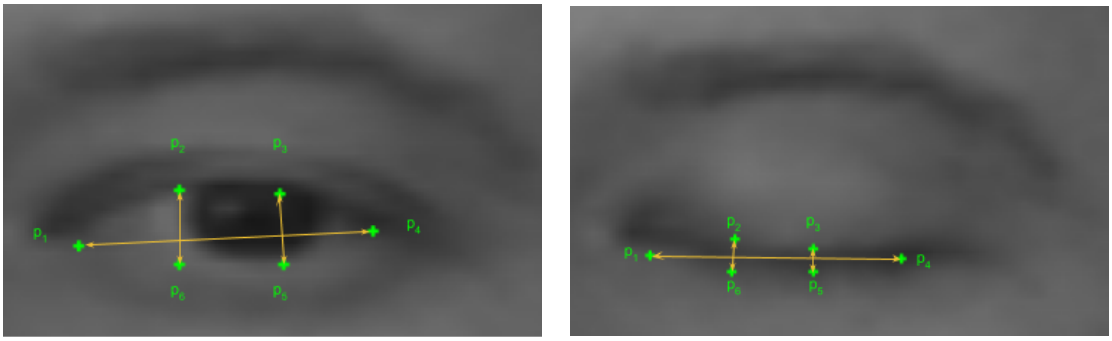


Figura 4.14: Imagens dos pontos utilizados.

A equação abaixo retorna o resultado do EAR para um olho:

$$EAR = \frac{\|p_2 - p_6\|}{2\|p_1 - p_4\|} \tag{4.1}$$

O sistema utiliza o valor médio calculado a partir de uma média aritmética com o EAR de cada olho. Caso este valor seja menor que o parâmetro determinado como limite no arquivo de configuração, o sistema infere que o usuário está com os olhos fechados.

4.1.6 Módulo de Recepção

O Módulo de Recepção é responsável por realizar uma chamada de procedimento remoto (RPC) que obtém os dados provenientes dos sensores conectados ao Raspberry Pi. Essa forma de comunicação é apresentada com mais detalhe na Seção 4.1.6.2. O módulo envia as informações obtidas para o Módulo de Severidade. O fluxo da informação pode ser visto na Figura 4.15. O módulo possui um parâmetro de *timeout* para cancelar a chamada caso o RabbitMQ ou o Módulo de Publicação estejam indisponíveis. Caso o tempo de resposta ultrapasse o parâmetro determinado, o método lança uma exceção que deve ser tratada pelo Módulo de Severidade. A mensagem a ser recebida contém uma 5-tupla no seguinte formato:

⟨Latitude; Longitude; Velocidade; Aceleração no eixo Y; Aceleração no eixo Z⟩,

sendo que tanto a Latitude quando a Longitude são representadas em graus, a velocidade em km/h e as acelerações em m/s^2. Um exemplo de uma mensagem válida seria:

$$\langle \text{-22,9077778; -43,1262885; 10; 0,13; 0,01} \rangle,$$

na qual a Latitude é igual a $-22,9077778^o$, a Longitude igual a $-43,1262885^o$, a velocidade igual a 10 km/h, a aceleração no eixo Y igual a $0,13$ m/s^2 e no eixo X igual a $0,01$ m/s^2.

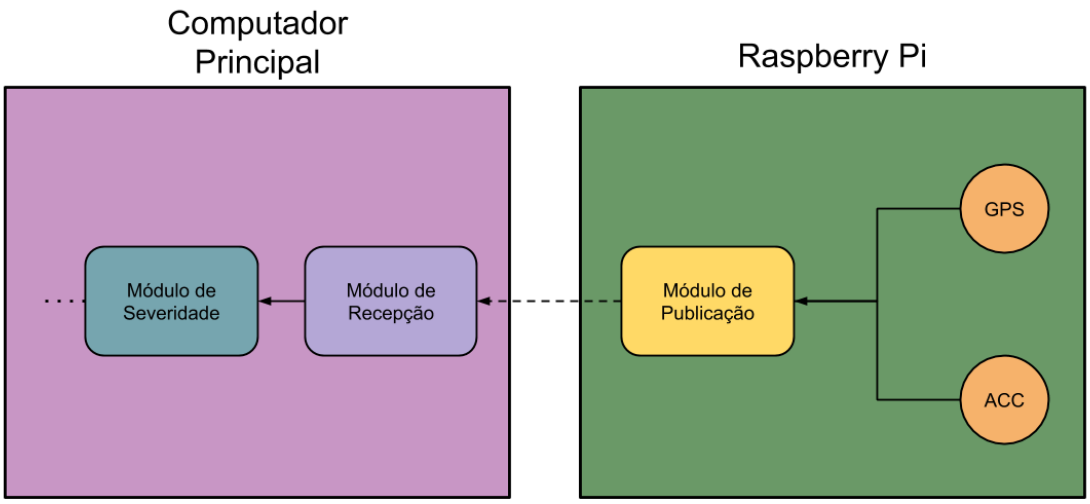


Figura 4.15: Fluxograma da obtenção dos dados dos sensores.

Uma vantagem desta notação é a possibilidade do aumento no número das fontes de dados. Caso um novo sensor seja adicionado ao sistema, basta acrescentar o dado proveniente do novo sensor na tupla. Por exemplo, caso fosse adicionado um sensor de chuva, um dado booleano poderia ser acrescentado na mensagem para informar se está (dado seria $true$) ou não chovendo (dado seria $false$). Dessa forma, a adição de um novo sensor se torna simples e não altera o funcionamento anterior do sistema.

4.1.6.1 RabbitMQ

O RabbitMQ é um *broker* de mensagens que suporta múltiplos protocolos de comunicação como o MQTT (*Message Queuing Telemetry Transport*) e o AMQP

(*Advanced Message Queuing Protocol*), por exemplo [29]. Com o uso de um *broker*, é possível criar filas, onde as mensagens publicadas pelos publicadores são armazenadas e ao serem consumidas pelos consumidores são excluídas. Desta forma, é possível implementar uma comunicação assíncrona, porque o publicador não necessita que sua mensagem publicada seja consumida pelo consumidor para continuar sua rotina. O RabbitMQ possui uma interface para a linguagem Python através da biblioteca Pika [30]. A arquitetura de comunicação utilizada foi a chamada de procedimento remoto (*Remote Procedure Call* - RPC) que será apresentada em mais detalhe na Seção 4.1.6.2. A escolha do RabbitMQ se deve ao fato de sua fácil implementação, escalabilidade e execução assíncrona.

4.1.6.2 Implementação da chamada de procedimento remoto

A chamada de procedimento remoto (RPC) torna possível uma rotina executar código em uma máquina remota [31]. O RabbitMQ é utilizado para a implementação desta forma de comunicação. O computador principal envia uma mensagem para a fila de requisição, atribuindo à esta mensagem as propriedades `correlation_id` e `reply_to`. Após o envio, aguarda a resposta observando a fila de resposta informada na propriedade *reply_to*. O Raspberry PI ao receber uma mensagem na fila de requisição executa a rotina e publica uma mensagem com id igual ao `correlation_id`, na fila `reply_to`. Por fim, o computador principal ao receber uma mensagem com id igual ao *correlation_id* fornecido retorna o resultado. Este procedimento está representado na Figura 4.16.

A implementação da comunicação por chamada de procedimento remoto simplifica a adição de novas fontes de dados provenientes de outras máquinas remotas. Para isso basta adicionar um novo publicador em uma nova fila e configurar o consumidor para se conectar nesta nova fila. Como este procedimento é assíncrono e executado em uma *thread* diferente da principal, a espera de mensagens não afeta o desempenho do sistema.

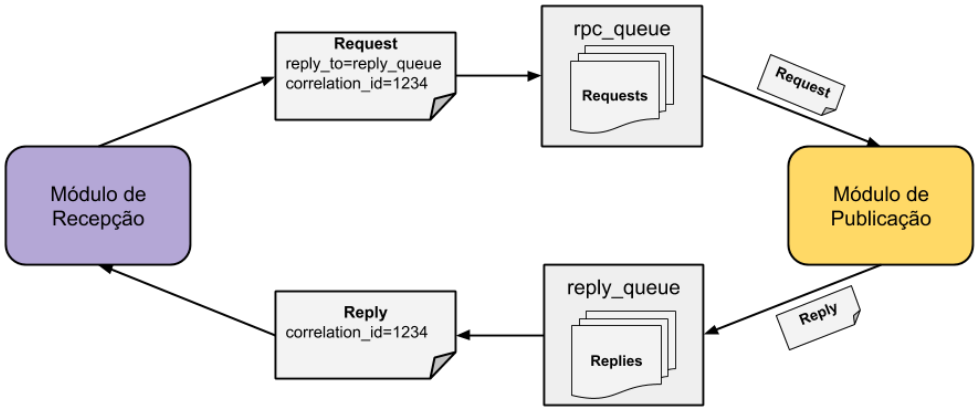


Figura 4.16: Diagrama de comunicação RPC.

4.1.7 Módulo de Publicação

O Módulo de Publicação é executado dentro do Raspberry Pi e tem o papel de responder às chamadas de procedimento remoto realizadas pelo computador principal. Portanto, ao receber uma chamada, ele coleta os dados dos sensores e monta a mensagem apresentada na Seção 4.1.6. A seguir será apresentado como a coleta dos dados de cada sensor é realizada.

4.1.7.1 Obtenção de dados do GPS

O GPS utilizado é o EVK-5 da empresa u-blox [32]. A comunicação serial entre o GPS e o Raspberry Pi é realizada via USB. O GPS envia uma mensagem com a estrutura abaixo e a descrição de cada campo pode ser vista na Tabela B.1 no Apêndice B.1 proveniente do *datasheet* do dispositivo:

$$\langle \$\text{GPRMC},\ldots,\text{status},\text{latitude},\text{N},\text{longitude},\ldots,\text{spd},\ldots \rangle$$

Portanto, para obter o dado do GPS, primeiramente é verificado se a mensagem é válida pelo atribulo "Status" (índice 2 na tabela). Caso o Status possua valor "A" (dados válidos) é feita a conversão dos campos de latitude e longitude para a unidade em graus, e a velocidade para km/h.

4.1.7.2 Obtenção dos dados do acelerômetro

O acelerômetro utilizado é o MPU-6050 [33]. A comunicação entre o Raspberry Pi e o acelerômetro é realizada com o protocolo I2C (*Inter-Integrated Circuit*).

A vantagem desse protocolo é a quantidade de entradas físicas necessárias. Apenas quatro fios são utilizados, sendo dois para o protocolo e dois para alimentação. Desta forma, poucas portas *GPIO* do Raspberry Pi são utilizadas, deixando espaço para sensores futuros. A conexão pode ser vista através da Figura 4.17.

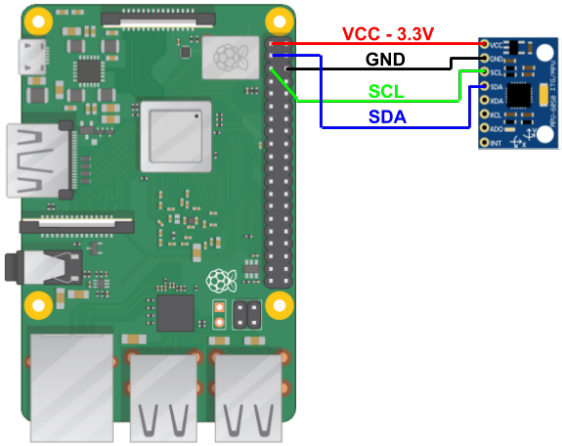


Figura 4.17: Conexão entre Raspberry PI e acelerômetro.

Após a conexão física entre o acelerômetro e o Raspberry Pi, é necessário realizar a configuração e instalação de pacotes. Este procedimento pode ser visto na Seção C.1 do Apêndice C.

Após a configuração, é possível testar se o acelerômetro está sendo detectado pelo Raspberry Pi com o comando abaixo. Como pode ser visto na Figura 4.18, o acelerômetro está sendo detectado com o endereço hexadecimal 0x68. Este endereço será utilizado para obter os dados dos registradores do acelerômetro.

```
sudo i2cdetect -y 1
```

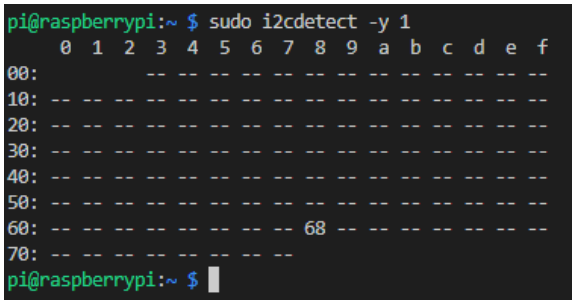


Figura 4.18: Saída do comando i2cdetect.

Para obter a informação da aceleração em um determinado eixo, é utilizada a função `read_byte_data(addr,cmd)` da biblioteca `smbus`, os argumentos dessa função

são o endereço do dispositivo (`addr`) e o endereço do registrador a ser lido (`cmd`). O endereço do dispositivo foi obtido anteriormente e os endereços dos registradores podem ser vistos na Tabela 4.1. Após a leitura do dado do registrador, basta convertê-lo para m/s^2.

Endereço (Hex)	**Nome**
6B	PWR_MGMT_1
3B	ACCEL_XOUT_H
3C	ACCEL_XOUT_L
3D	ACCEL_YOUT_H
3E	ACCEL_YOUT_L
3F	ACCEL_ZOUT_H
40	ACCEL_ZOUT_L

Tabela 4.1: Tabela dos registradores do MPU-6050.

4.1.8 Módulo de Notificação

Este módulo é implementado de maneira similar ao Módulo de Publicação, porém sem a utilização da chamada de procedimento remoto. O Módulo de Notificação utiliza uma arquitetura simples de Publicador e Consumidor de mensagens. Ele possui a funcionalidade de ser o publicador para as mensagens de estado do motorista. Ao receber o estado do Módulo Principal, o módulo repassa este estado através de uma mensagem para o Módulo de Alerta que possui a funcionalidade de consumidor e será apresentado na Seção 4.1.9. É utilizada uma fila separada para as mensagens de alerta, as mensagens contêm apenas uma *string true* para caso o estado de distração seja detectado e *false* para o contrário. A vantagem de utilizar esta arquitetura é a possibilidade de conectar mais de um consumidor a esta fila. Desta forma, diferentes módulos podem consumir esta informação de forma assíncrona. Um exemplo seria, ao detectar uma mensagem de desatenção, ao mesmo tempo que um módulo pode alertar o usuário, outro módulo poderia bloquear o uso do celular.

4.1.9 Módulo de Alerta

O Módulo de Alerta possui como principal funcionalidade alertar o usuário caso seja inferido pelo sistema que o mesmo encontra-se distraído ou sonolento.

Desta forma, caso o usuário possua este estado, o módulo é responsável por recuperar a atenção do motorista. Este módulo funciona como um consumidor em uma arquitetura de Publicador e Consumidor. Ele é executado dentro do Raspberry Pi e funciona de forma independente do Módulo de Publicação anteriormente apresentado na Seção 4.1.7 Ao receber uma mensagem, o módulo trata o valor recebido, para o caso positivo *true* (desatento), o módulo liga um *led* e emite um som para capturar a atenção do usuário. O *led* e o som apenas são desligados quando o módulo recebe uma mensagem com o valor *false*, indicando que o usuário se encontra atento. O fluxograma da comunicação entre o Módulo de Notificação e o Módulo de Alerta é apresentado a seguir pela Figura 4.19.

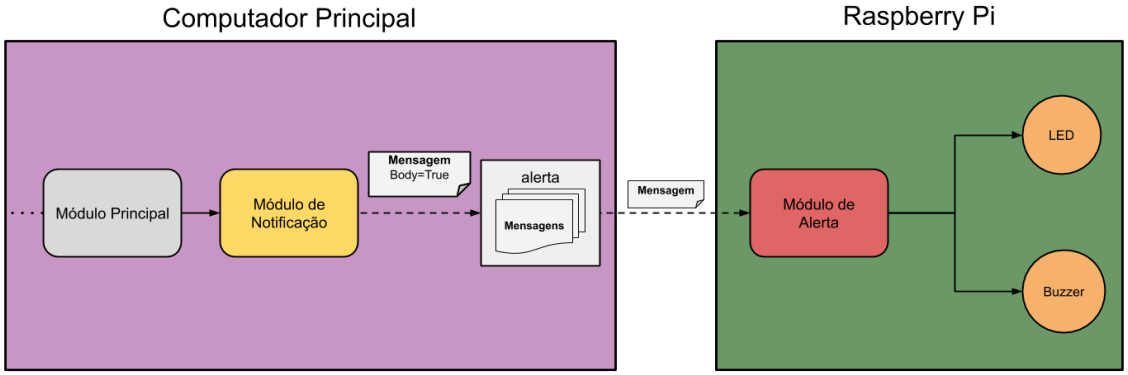


Figura 4.19: Fluxograma de comunicação entre os módulos de Notificação e Alerta.

Capítulo 5

Experimentos e Resultados

Este capítulo contém os testes realizados durante o desenvolvimento do sistema. Os testes têm como finalidade validar o funcionamento individual e integrado dos módulos e suas funcionalidades apresentadas nos capítulos anteriores. A vantagem da realização dos testes individuais é que em caso de erro, a identificação do problema se torna mais simples por causa do escopo reduzido.

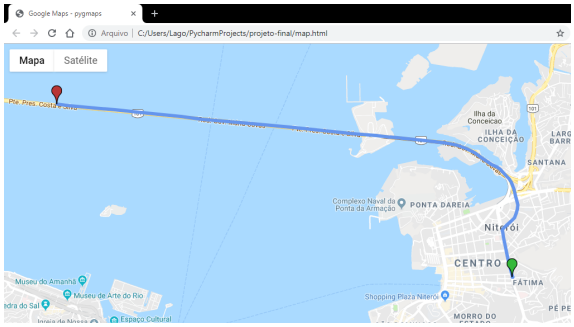
5.1 Testes preliminares

Este teste tem como objetivo verificar a aquisição dos dados provenientes do GPS e do acelerômetro.

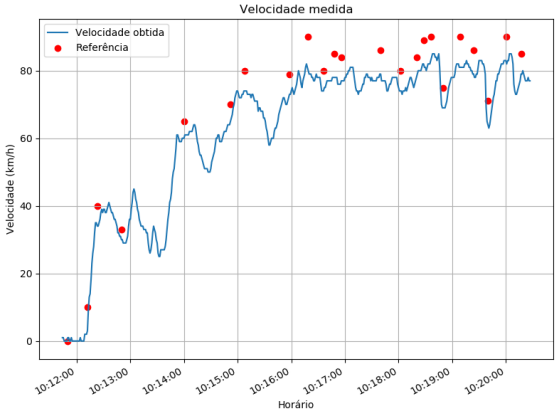
5.1.1 GPS

Apesar da posição ser verificada durante testes de bancada, para analisar a velocidade retornada pelo GPS é necessário realizar o teste em um veículo em movimento. Para este teste, o GPS foi conectado diretamente em um notebook e foi criado um *script* em *Python* que armazena a cada segundo a latitude, longitude e velocidade medida pelo GPS. A velocidade do veículo exibida pelo velocímetro é anotada periodicamente, de forma manual, a fim de servir como referência. O conjunto de Figuras 5.1 é gerado com os dados armazenados no arquivo durante o teste. A Figura 5.1a foi criada a partir do mesmo *script* que é utilizado no sistema projetado, a marcação verde delimita o início do trajeto, enquanto que a vermelha o fim. A Figura 5.1b mostra a velocidade do veículo durante o percurso. Os pontos

vermelhos são as velocidades anotadas do velocímetro, e a linha azul a velocidade obtida pelo GPS.



(a) Trajetória do veículo representada pela linha vermelha.



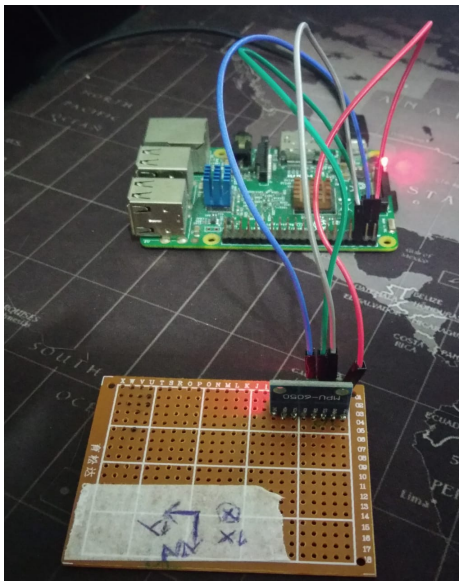
(b) Gráfico da velocidade obtida pelo GPS em azul e referência em vermelho.

Figura 5.1: Resultados do teste do GPS.

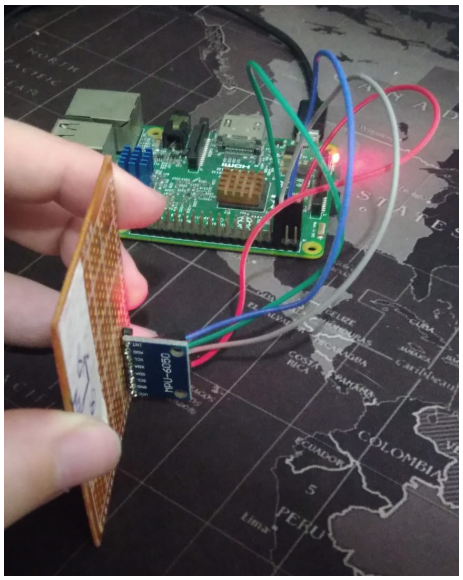
A velocidade do veículo medida pelo GPS apresenta um erro significativo em velocidades mais elevadas. Porém o GPS consegue acompanhar com eficiência momentos de aumento e redução de velocidade. O erro na velocidade não é um problema para a aplicação porque não é necessário saber a velocidade exata do veículo, basta uma estimativa. Além disso, nota-se que a posição do veículo não apresenta nenhum problema, sendo possível traçar a trajetória correta do veículo.

5.1.2 Acelerômetro

O teste do acelerômetro deve ser realizado após a configuração descrita no Apêndice C e Seção C.1. Este teste tem o propósito de checar o correto funcionamento do acelerômetro utilizado e sua devida comunicação com o Raspberry Pi. Para verificar se os valores retornados condizem com a realidade, é necessário avaliar os resultados com o acelerômetro em diferentes posições. Para o teste a seguir, duas configurações foram utilizadas, o acelerômetro em repouso e inclinado. Desta forma, a alteração da inclinação deve modificar o valor exibido para cada eixo devido à força gravitacional.



(a) Configuração A: acelerômetro em repouso.



(b) Configuração B: acelerômetro inclinado.

Figura 5.2: Configurações para o teste do acelerômetro.

Configuração	**Eixo X**	**Eixo Y**	**Eixo Z**
A	1.019	0.01	0.082
B	0.048	-1.015	0.052

Tabela 5.1: Resultado das medições para a configuração A e B.

O Conjunto de Figuras 5.2 mostra as configurações A e B utilizadas para o teste do acelerômetro. E a Tabela 5.1 exibe os valores obtidos de aceleração para cada configuração. Percebe-se que para a primeira configuração, o valor da aceleração no eixo x permanece em torno de 1 m/s^2, enquanto que nos demais eixos permanecem próximos de 0 m/s^2. Ao trocar para a segunda configuração, o valor do eixo y passa a ser próximo ao valor da aceleração gravitacional enquanto que nos outros eixos fica nulo.

5.2 Testes do Módulo de Face

O objetivo desta seção é apresentar os testes realizados que envolvam o Módulo de Face.

5.2.1 Sem integração

O primeiro teste realizado tem como objetivo verificar o funcionamento do Módulo de Face operando sem integração com os sensores. Portanto, o módulo deve utilizar os valores padrões para os limites de rotação transversal e longitudinal. Quando o ângulo medido ultrapassa seu limite a distração é detectada. Outra característica capaz de ser notada é o funcionamento dos contadores, pois a distração não é atribuída assim que um limite é ultrapassado, mas sim após um determinado período de tempo.

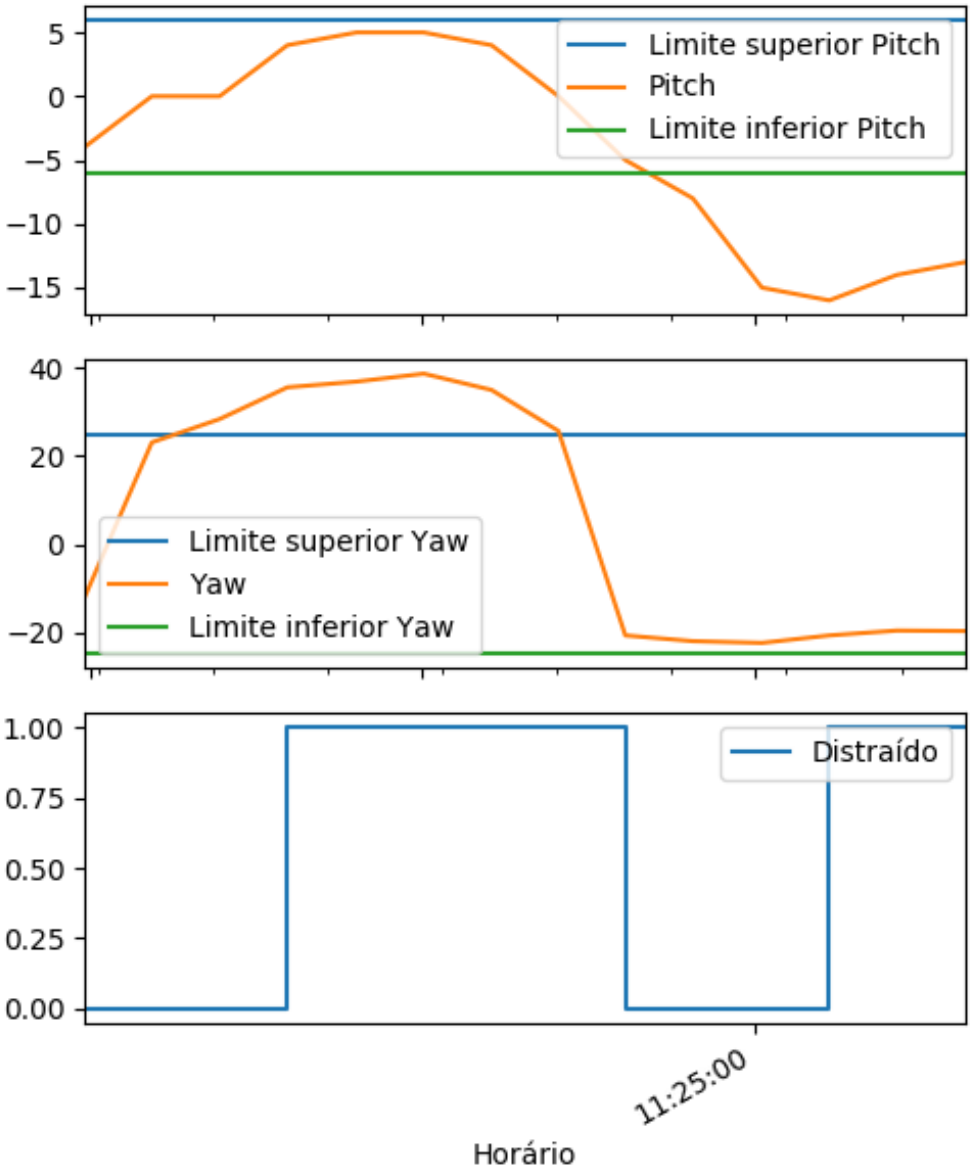


Figura 5.3: Gráfico dos valores de Pitch e Yaw ao longo do tempo e distração detectada.

(a) Olhando para a direita. (b) Olhando para baixo.

Figura 5.4: Teste do módulo de face utilizando parâmetros padrões.

A Figura 5.3 mostra o funcionamento do Módulo de Face sem a utilização dos sensores adicionais. Portanto, os valores padrões de limites estão sendo utilizados. Para este teste foi utilizado um limite de 25° no eixo longitudinal (*Yaw*) e um limite de 5° para o eixo transversal (*Pitch*). Portanto, quando o *Pitch* ou *Yaw* ultrapassam seus limites definidos é determinado que o motorista está distraído. Os momentos de desatenção podem ser vistos com as Figuras 5.4a e 5.4b.

5.2.2 Com integração

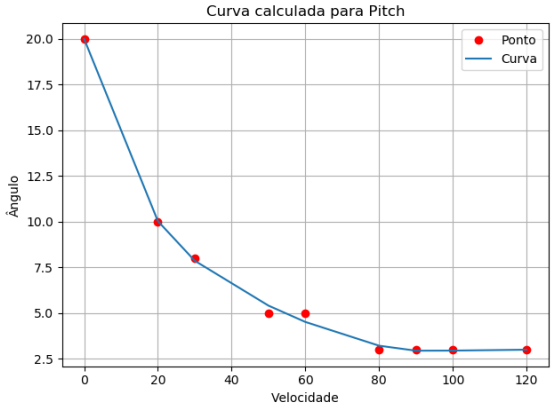
A seguir são apresentados os testes que possuem a integração do Módulo de Face com os sensores. Portanto, os valores dos limites de rotação longitudinal e transversal devem variar de acordo com os dados dos sensores.

5.2.2.1 Apenas velocidade

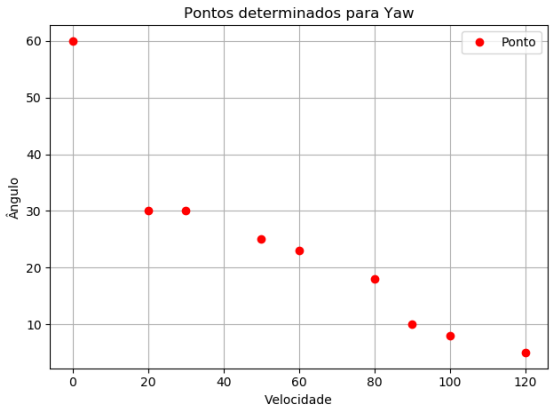
Para o teste a seguir, o Módulo de Publicação foi modificado com a finalidade de aumentar a velocidade enviada em 1 km/h a cada segundo. Desta forma, é possível testar o comportamento do Módulo de Face com a integração dos parâmetros de velocidade do Módulo de Severidade de forma segura e sem a necessidade de realizá-lo em um veículo. Espera-se que neste teste seja possível verificar a integração do Módulo de Severidade com o Módulo de Recepção e que os parâmetros utilizados como limites de rotação sejam influenciados pela velocidade.

Os pontos utilizados no arquivo de configuração e as curvas de *ângulo limite x velocidade* geradas a partir deles estão representados no conjunto de figuras abaixo:

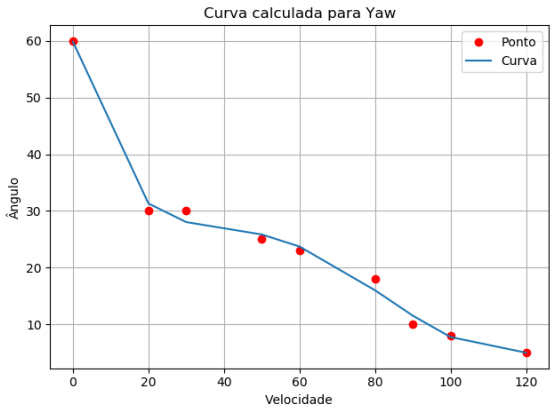
(a) Pontos para o eixo transversal.



(b) Curva gerada a partir dos pontos configurados.



(c) Pontos para o eixo longitudinal.



(d) Curva gerada a partir dos pontos configurados.

Figura 5.5: Parâmetros utilizados durante o teste.

O Conjunto de Figuras 5.7 mostra o resultado do teste de integração proposto. A Figura 5.6 exemplifica a variação dos limites definidos pelo Módulo de Severidade com o aumento da velocidade, além do valor calculado para o *Pitch* e *Yaw*. Também é possível verificar a classificação de distração do usuário, quando um dos ângulos supera o seu limite o sistema detecta a distração. As Figuras 5.7a e 5.7b mostram situações em que o sistema classificou o usuário como distraído durante o teste e é possível saber o valor da velocidade através do campo *"Spd"*. Com estas figuras, nota-se a mudança dos limites determinados pelo Módulo de Severidade de acordo com a velocidade, conforme as curvas de *ângulo limite x velocidade* apresentadas anteriormente pelas Figuras 5.5b e 5.5d. Para valores mais altos de velocidade os limites atribuídos são menores em relação ao teste anterior, portanto, o sistema infere

a direção do olhar com ângulos menores. Enquanto que para valores mais baixos os limites são consideravelmente maiores, permitindo ao motorista um movimento mais amplo.

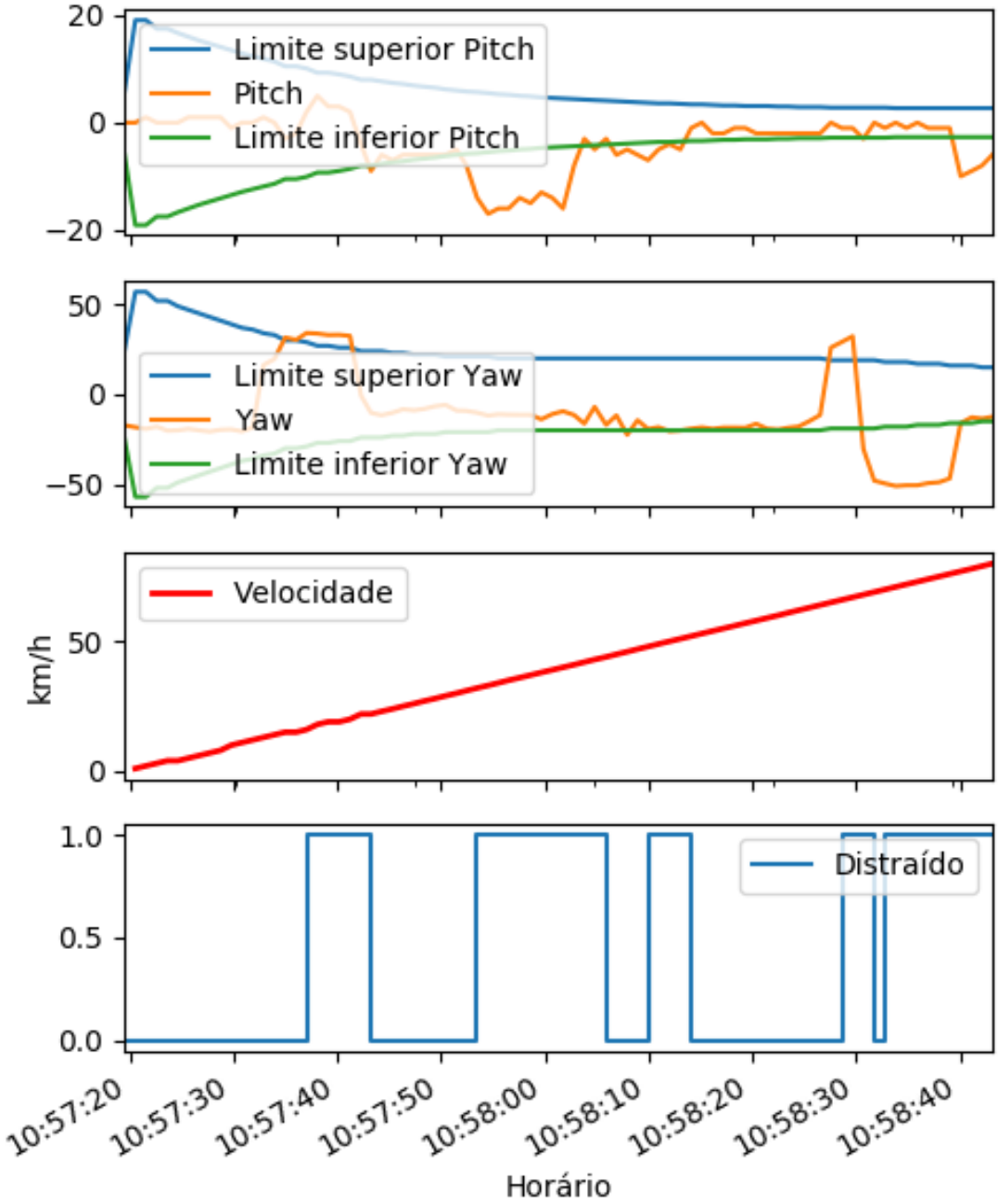


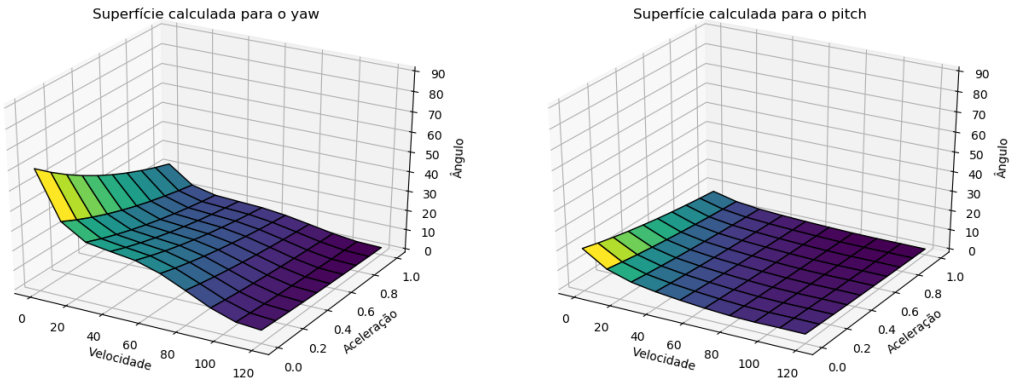
Figura 5.6: Gráfico dos valores de Pitch e Yaw ao longo do tempo.

(a) Olhando para a direita. (b) Olhando para baixo.

Figura 5.7: Teste do módulo de face com integração de velocidade.

5.2.2.2 Velocidade e acelerômetro

O teste a seguir foi realizado com o mesmo procedimento do teste anterior. A velocidade é incrementada em 5 km/h e a aceleração em 0.1 m/s^2 a cada segundo neste teste. As Figuras 5.8a e 5.8b mostram a superfície gerada através dos parâmetros informados no arquivo de configuração.



(a) Superfície que delimita os limites para o *yaw*. (b) Superfície que delimita os limites para o *pitch*.

Figura 5.8: Superfícies utilizadas no teste.

Na Figura 5.9 é possível verificar que os limites determinados são influenciados tanto pela velocidade como aceleração. Portanto, os limites decrescem com uma velocidade maior do que a obtida no teste anterior apenas com a velocidade. O momento de distração detectado pelo sistema pode ser visto através da Figura 5.10

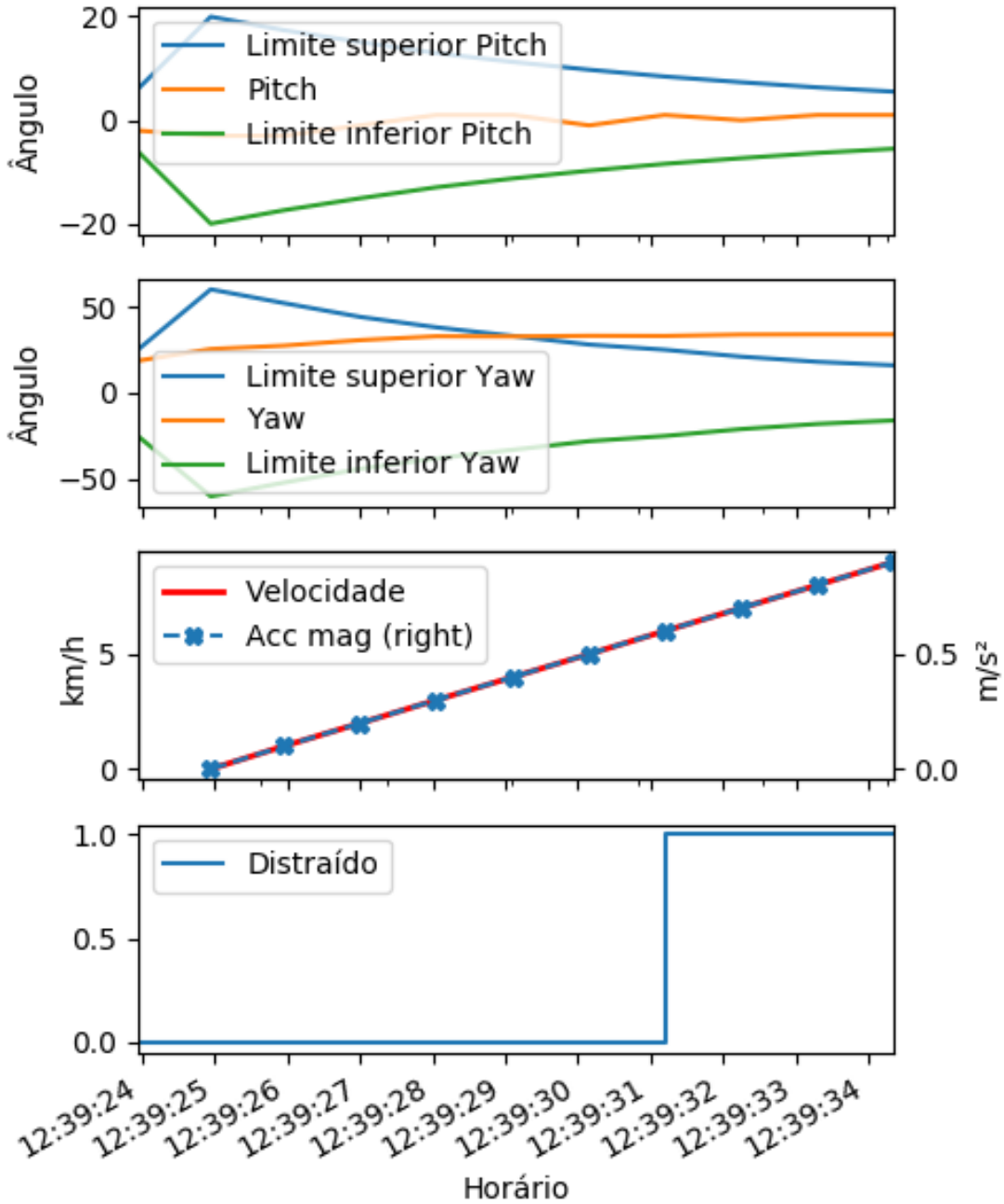
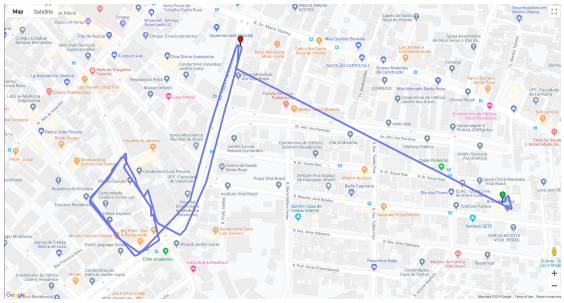


Figura 5.9: Gráfico dos valores de Pitch e Yaw ao longo do tempo com influência da velocidade e aceleração.

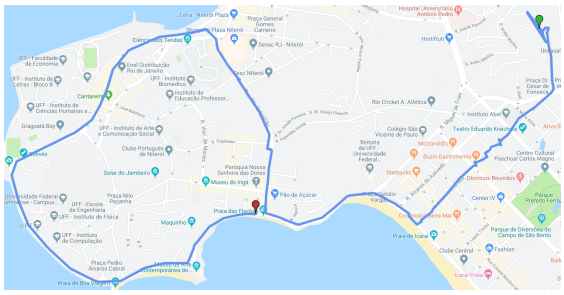
Figura 5.10: Olhando para direita com influência da velocidade e acelerômetro.

5.3 Teste em campo

Nesta seção é apresentado o resultado dos testes realizados em campo. Este tipo de teste tem como objetivo verificar o funcionamento do sistema em seu ambiente de operação. Foram realizados dois testes, um no período da tarde e outro no entardecer. O sistema foi instalado em um veículo e um motorista dirigiu na cidade sendo analisado pelo sistema. As Figuras 5.11a e 5.11b apresentam os resultados dos mapas gerados pelo sistema após os testes.



(a) Percurso realizado de tarde.



(b) Percurso realizado no entardecer.

Figura 5.11: Rota registrada pelo sistema. Inicio na marcação verde e fim na vermelha.

A Figura 5.12 mostra o resultado obtido para os limites e os valores de *Pitch* e *Yaw* durante o teste em campo. Nos momentos em que o valor ultrapassou o limite superior, inferior ou detectou olhos fechados, o sistema detectou a desatenção do motorista, as Figuras 5.14a e 5.14b ilustram momentos de desatenção do motorista.

Além disso, é possível verificar que quando houve falha na obtenção da velocidade do GPS, o sistema utilizou os valores padrões para os limites de rotação da face.

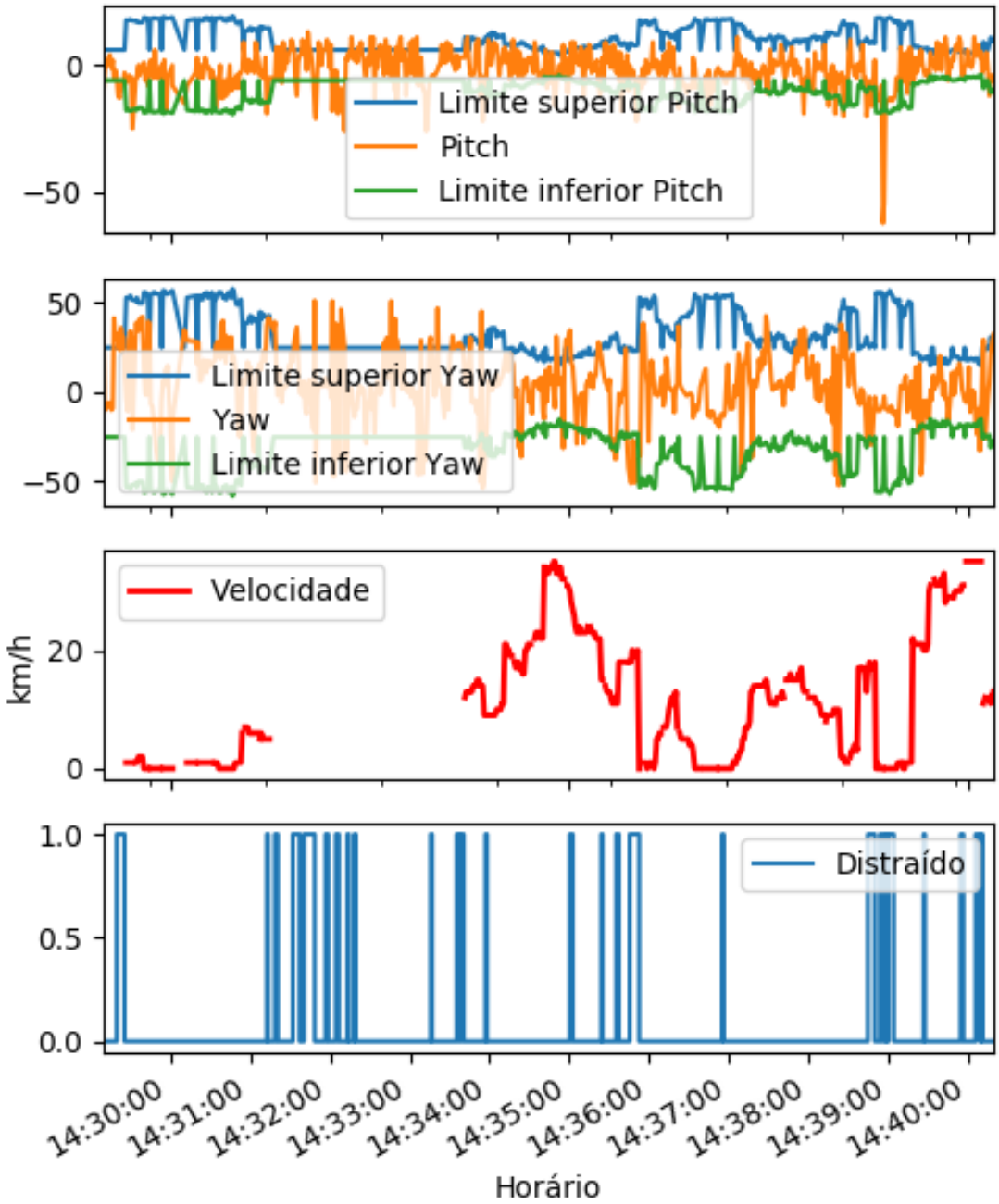


Figura 5.12: Resultado dos ângulos de rotação ao longo do tempo.

Durante o teste, o sistema detectou o fechamento dos olhos do motorista conforme pode ser visto na Figura 5.14b. O gráfico na Figura 5.13 mostra os valores medidos de EAR e o limite utilizado. Neste gráfico é possível notar que apenas quando o valor de EAR ficou abaixo do limite por um determinado número de

quadros consecutivos, o sistema inferiu o fechamento dos olhos. Quando há quedas bruscas no valor por um instante o sistema não detecta a distração por conta da implementação dos contadores.

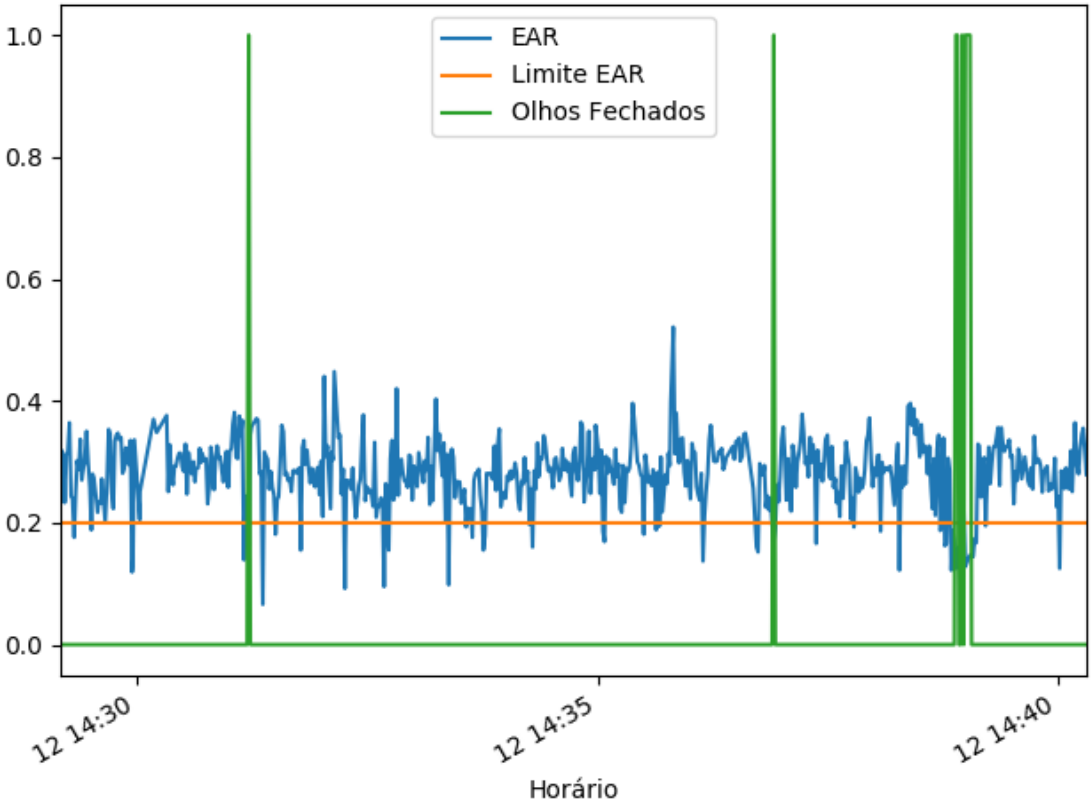


Figura 5.13: Valores e eye aspect ratio medidos pelo sistema.

(a) Sistema detectando uma distração.

(b) Sistema detectando fechamento dos olhos.

Figura 5.14

Figura 5.15: Sistema detectando uma falsa distração.

Uma falha percebida durante os testes é que ao acompanhar uma curva com o olhar, o sistema detecta uma falsa distração e portanto alerta o usuário como pode ser visto na Figura 5.15. Na mesma figura pode ser visto que ao perder o sinal de GPS o sistema continua operando utilizando os valores padrões para os limites de rotação longitudinal e transversal.

(a) Falha na detecção do rosto.

(b) Falha do sistema ao detectar pontos faciais devido à falta de iluminação.

Figura 5.16

Quando o motorista vira o rosto a ponto do sistema não conseguir detectar

os pontos de interesse na face, o sistema não detecta a desatenção do motorista, esta ocorrência pode ser verificada na Figura 5.16a. Este problema se tornou mais frequente durante o teste realizado no entardecer, a iluminação do ambiente também influencia o correto funcionamento, quando há falta de iluminação, o sistema apresentou dificuldades para detectar os pontos de interesse na face do motorista conforme pode ser visto na Figura 5.16b.

Capítulo 6

Conclusão

6.1 Conclusão

Conforme foi apresentado no trabalho, o objetivo foi construir um sistema capaz de detectar e alertar momentos em que um motorista se encontra desatento. Foram apresentados trabalhos que solucionam o problema de inferência de distração. Porém nenhum deles realiza a análise utilizando dados complementares e em tempo real. O sistema desenvolvido neste trabalho se enquadrou nos campos visual, comportamental e fisiológico por utilizar dados de diferentes fontes. O projeto atendeu com sucesso ao objetivo proposto, ele detectou momentos de desatenção do usuário e alertou em tempo real, além disso, suas críticas foram realizadas utilizando parâmetros variáveis em relação ao contexto do veículo, o que torna a análise mais flexível para o motorista. A característica modular do sistema se provou eficiente para o desenvolvimento e a manutenção durante a realização do trabalho, além de garantir o correto funcionamento do sistema quando algum módulo externo perde a conexão.

Para momentos em que o rosto do motorista está completamente visível e em condições boas de iluminação, o sistema sempre detecta a distração do motorista. Porém em momentos que o motorista desvia apenas o olhar e não o rosto inteiro o sistema não classifica como uma distração. Além disso, o sistema apresentou falso positivo ao motorista acompanhar curvas longas com o olhar.

Os resultados apresentados mostram um bom desempenho e funcionamento, além de apresentar pequenas melhorias que podem ser realizadas em trabalhos fu-

turos.

6.2 Trabalhos futuros

No Capítulo 5 foram apresentadas alguns pontos de melhoria que foram detectados durante o teste em campo. Como por exemplo a dificuldade de detecção dos pontos de interesse da face do usuário em ambientes com pouca iluminação e o falso positivo do sistema ao motorista acompanhar uma curva com o olhar. Para o segundo problema, pode ser utilizado um giroscópio para verificar se o motorista está realizando uma curva e não identificar como distração caso o motorista esteja olhando para a direção da curva. Além destas melhorias, a característica modular do sistema permite a adição de novos sensores, portanto podem existir novos trabalhos ao adicionar diferentes sensores com a finalidade de detectar outras fontes de distração como auditiva por exemplo.

Outra linha de trabalho que pode ser realizado é a utilização de aprendizado de máquina para a análise dos dados obtidos pelo sistema. Desta forma, um classificador poderia decidir se o motorista está distraído ou não, sem a necessidade da criação de uma curva empírica.

Referências Bibliográficas

[1] "Evoluçao do Número de Acessos Móveis no Tempo", http://www.anatel.gov.br/dados/acessos-telefonia-movel, Acessado em: 2019-03-27.

[2] CAIRD, J. K., JOHNSTON, K. A., WILLNESS, C. R., *et al.*, "A meta-analysis of the effects of texting on driving", *Accident Analysis & Prevention*, v. 71, pp. 311–318, 2014.

[3] "Usar celular na direção é terceira causa de mortes no trânsito do Brasil", http://g1.globo.com/jornal-hoje/noticia/2017/09/usar-celular-na-direcao-e-terceira-causa-de-mortes-no-transito-do-brasil.html, Acessado em: 2019-03-27.

[4] *Distracted Driving in Fatal Crashes, 2017*, Report, 2017.

[5] RANNEY, T. A., GARROTT, W. R., GOODMAN, M. J., *NHTSA driver distraction research: Past, present, and future*, Report, SAE Technical Paper, 2001.

[6] RANNEY, T. A., GARROTT, W. R., GOODMAN, M. J., *NHTSA driver distraction research: Past, present, and future*, Report, SAE Technical Paper, 2001.

[7] PHILIP, P., SAGASPE, P., MOORE, N., *et al.*, "Fatigue, sleep restriction and driving performance", *Accident Analysis & Prevention*, v. 37, n. 3, pp. 473 – 478, 2005.

[8] "Audi - Rest recommendation system", https://www.audi-mediacenter.com/en/foray-into-the-worlds-largest-market-segment-the-audi-a3-sedan-and-s3-s driver-assistance-systems-3350, 2013, Acessado em: 2019-09-15.

[9] "BMW - Active driving assistant", https://www.bmw.ca/en/topics/experience/connected-drive/bmw-connecteddrive-driver-assistance.html, Acessado em: 2019-09-28.

[10] "Volkswagen - Driver Alert System", https://www.volkswagen-newsroom.com/en/driver-alert-system-3932, Acessado em: 2019-09-28.

[11] "Ford - Driver Alert System", https://owner.ford.com/how-tos/vehicle-features/safety/driver-alert-system.html, Acessado em: 2019-09-28.

[12] ALGHOWINEM, S., GOECKE, R., WAGNER, M., *et al.*, "Head pose and movement analysis as an indicator of depression". In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pp. 283–288, IEEE, 2013.

[13] RAY, S. J., TEIZER, J., "Coarse head pose estimation of construction equipment operators to formulate dynamic blind spots", *Advanced Engineering Informatics*, v. 26, n. 1, pp. 117–130, 2012.

[14] FRIDMAN, L., LEE, J., REIMER, B., *et al.*, "'Owl' and 'Lizard': patterns of head pose and eye pose in driver gaze classification", *IET Computer Vision*, v. 10, pp. 308–314(6), June 2016.

[15] "Jornal O Povo entrevista com Victor Pavarino: Dificuldade em fiscalizar e punir", https://www.unifor.br/web/osv/como-o-celular-prejudica-a-ateno-ao-trnsito, 2018, Acessado em: 2019-09-15.

[16] WANG, Y.-K., CHEN, S.-A., LIN, C.-T., "An EEG-based brain–computer interface for dual task driving detection", *Neurocomputing*, v. 129, pp. 85–93, 2014.

[17] KHANDAKAR, A., CHOWDHURY, M. E., AHMED, R., *et al.*, "Portable system for monitoring and controlling driver behavior and the use of a mobile phone while driving", *Sensors*, v. 19, n. 7, pp. 1563, 2019.

[18] CONAMA, "Resolução CONAMA nº 354 de 13/12/2004 - Art. 3º", 2004.

[19] ALAM, L., HOQUE, M. M., "Real-Time Distraction Detection Based on Driver's Visual Features". In: *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–6, IEEE, 2019.

[20] CRAYE, C., RASHWAN, A., KAMEL, M. S., *et al.*, "A multi-modal driver fatigue and distraction assessment system", *International Journal of Intelligent Transportation Systems Research*, v. 14, n. 3, pp. 173–194, 2016.

[21] KING, D. E., "Dlib-ml: A Machine Learning Toolkit", *Journal of Machine Learning Research*, v. 10, pp. 1755–1758, 2009.

[22] KAZEMI, V., SULLIVAN, J., "One millisecond face alignment with an ensemble of regression trees". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1867–1874, 2014.

[23] ZAFEIRIOU, S., TZIMIROPOULOS, G., PANTIC, M., "The 300 videos in the wild (300-VW) facial landmark tracking in-the-wild challenge". In: *ICCV Workshop*, v. 32, p. 73, 2015.

[24] BRADSKI, G., "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, , 2000.

[25] "Biblioteca Skyfield", `https://rhodesmill.org/skyfield/`, Acessado em: 2019-08-10.

[26] SOUKUPOVA, T., CECH, J., "Eye blink detection using facial landmarks". In: *21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia*, 2016.

[27] "Modelo rosto humano 3D", `https://www.turbosquid.com/FullPreview/Index.cfm/ID/346686`, Acessado em: 2019-08-14.

[28] "Blender", `https://www.blender.org/`, Acessado em: 2019-08-14.

[29] "Documentação RabbitMQ", `https://www.rabbitmq.com/documentation.html`, Acessado em: 2019-08-04.

[30] "Biblioteca Pika", `https://pika.readthedocs.io/en/stable/`, Acessado em: 2019-08-04.

[31] BIRRELL, ANDREW D.; NELSON, B. J., "Implementing remote procedure calls", *ACM Transactions on Computer Systems*, v. 2, pp. 39–59, 1984.

[32] "U-Blox", `https://www.u-blox.com/en`, Acessado em: 2019-08-14.

[33] "Datasheet do acelerômetro MPU-6050", `https://www.alldatasheet.com/datasheet-pdf/pdf/517744/ETC1/MPU-6050.html`, Acessado em: 2019-08-18.

[34] ROWEIS, S., "Levenberg-marquardt optimization", *Notes, University Of Toronto*, , 1996.

[35] "Biblioteca smbus", `http://wiki.erazor-zone.de/wiki:linux:python:smbus:doc`, Acessado em: 2019-08-18.

Apêndice A

Cálculo dos ângulos

A.1 Calculo dos ângulos

Como inicialmente apresentado na Seção 4.1.5.1. A Figura A.1 ilustra os três sistemas de coordenadas em questão.

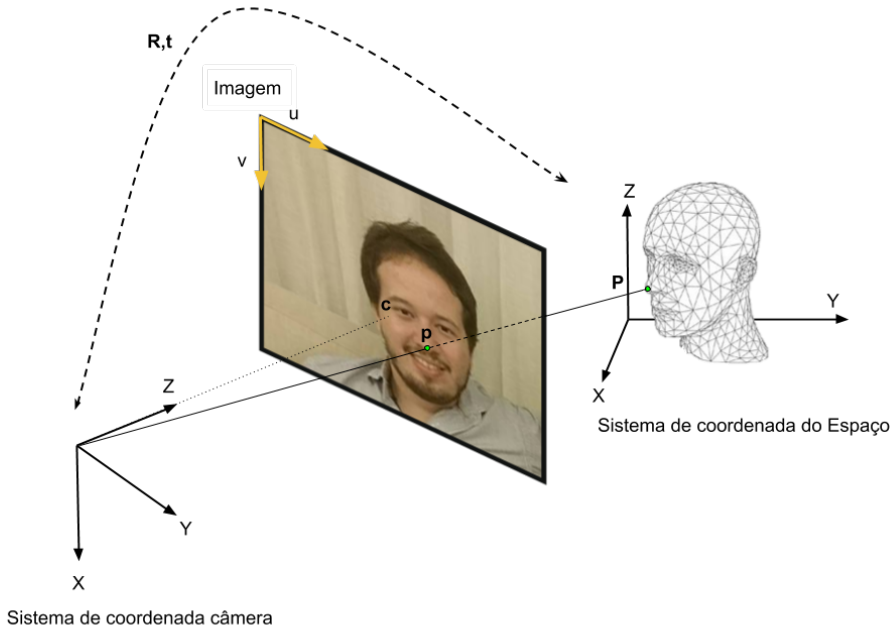


Figura A.1: Ilustração dos sistemas de coordenadas utilizados.

O vetor **V** contém os pontos de interesse da face no modelo 3D como pode ser visto na Figura A.1 e foi obtido através do processo exemplificado na Seção 4.1.5.1.

O ponto P na figura é representado na primeira linha deste vetor.

$$\mathbf{V} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 5.68417 & 3.9121 \\ 12.4519 & 8.83946 & 19.6382 \\ -12.4519 & 8.83946 & 19.6382 \\ 9.07492 & 7.6968 & 6.7581 \\ 3.06506 & 8.25098 & 4.8036 \\ -3.06506 & 8.25098 & 4.8036 \\ -9.07492 & 7.6968 & 6.7581 \\ 8.27412 & 5.24929 & 7.2309 \\ 3.37092 & 5.42769 & 6.5221 \\ -3.37092 & 5.42769 & 6.5221 \\ -8.27412 & 5.24929 & 7.2309 \\ 2.93488 & -1.72152 & 4.018 \\ -2.93488 & -1.72152 & 4.018 \\ 4.6196 & -6.2328 & 5.5864 \\ -4.6196 & -6.2328 & 5.5864 \\ 0 & -8.2701 & 4.1881 \\ 0 & -12.9503 & 4.0926 \end{bmatrix} \tag{A.1}$$

Como objetivo, é necessário obter a matriz de rotação do sistema de coordenada do espaço para o sistema de coordenada da câmera. A Equação A.2 representa a transformação do ponto **P** no espaço para o ponto **p** no sistema de coordenadas da imagem.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t} \tag{A.2}$$

Porém, os parâmetros da câmera representados pela matriz **A** devem ser considerados, sendo que f_x e f_y são as distâncias focais da câmera, e (c_x,c_y) é o centro óptico. A distorção radial é desconsiderada.

$$\mathbf{A} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

A Equação A.4 representa o problema a ser resolvido utilizando a otimização de Levenberg-Marquardt. [34]

$$z.\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{A.4}$$

O ponto (u,v) é um ponto de interesse obtido através da biblioteca Dlib, o ponto (X,Y,Z) é um ponto arbitrário no espaço referente ao mesmo ponto (u,v). r_{ij} e (t_x,t_y,t_z) são as variáveis desejadas. Desta forma, a matriz de rotação e translado é obtida.

A otimização de Levenberg-Marquardt pode ser utilizada através da função *solvePnP()* com a flag SOLVEPNP_ITERATIVE. Com esta função é obtido o vetor de rotação e o vetor de translado. A função *Rodrigues()* da biblioteca OpenCv transforma um vetor de rotação em uma matriz de rotação. Portanto utilizando esta função com a saída da *solvePnp()* é possível obter a matriz de rotação desejada.

Apêndice B

Informações complementares

B.1 GPS

Tabela B.1: Tabela do protocolo utilizado no GPS

Índice	Exemplo	Nome	Unidade	Descrição
0	$GPRMC	$GPRMC	-	Cabeçalho do protocolo RMC
1	083559.00	hhmmss.ss	-	Timestamp
2	A	Status	-	A = Dados válidos / V = Alerta Receptor de navegação
3	4717.11437	Latitude	-	Latitude, Graus + minutos
4	N	N	-	Indicador N/S, hemisfério N=Norte , S=Sul
5	00833.91522	Longitude	-	Longitude, Graus + minutos
6	E	E	-	Indicador E/W, E=East , W=West
7	0.004	Spd	nós	Velocidade
8	77.52	Cog	ângulo	Curso sobre o solo
9	091202	date	-	Dia, mês, ano
10	-	mv	ângulo	Valor variação magnética
11	-	mvE	-	Indicador E/W variação magnética
12	-	mode	-	Modo indicador
13	*57	cs	-	Verificador
14	-	<CR><LF>	-	Código de fim de linha

Apêndice C

Configurações

C.1 Configuração do Raspberry Pi

O barramento SPI *Serial Peripheral Interface* e o protocolo I2C do Raspberry Pi precisam ser configurados e ativados através de um menu de configuração (Figura C.1) que pode ser aberto com o seguinte comando:

```
sudo raspi-config
```

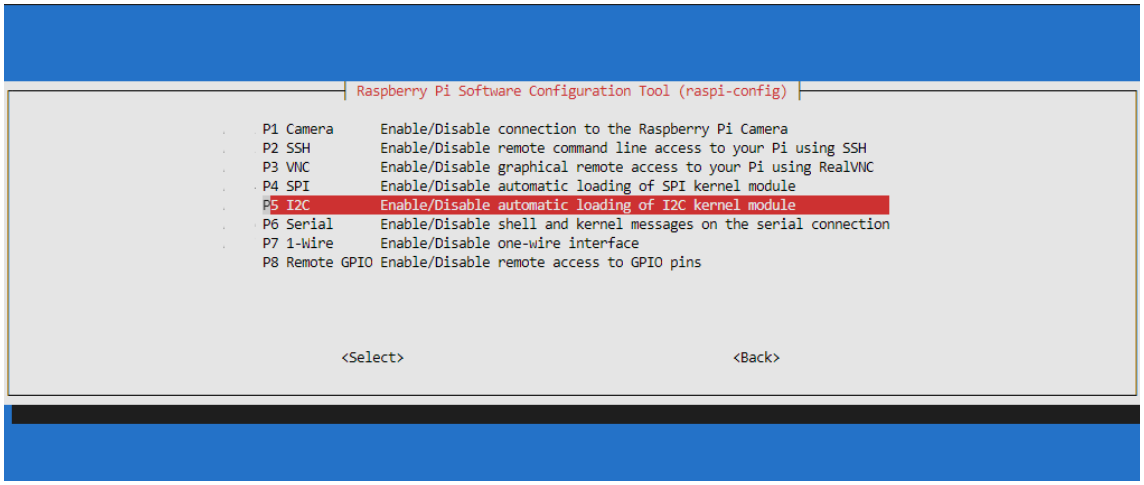


Figura C.1: Menu de configuração.

Após a configuração, é preciso instalar alguns pacotes no Raspberry Pi. O pacote `i2c-tools` é necessário para obtenção do endereço do dispositivo e a biblioteca `python-smbus` [35] é usada para obtenção dos dados do acelerômetro. A instalação dos dois pacotes requer a execução do seguinte comando:

```
sudo apt-get install i2c-tools python-smbus
```